A TRIDENT SCHOLAR PROJECT REPORT

NO. 271

ERROR STATISTICS OF TIME-DELAY EMBEDDING PREDICTION ON CHAOTIC TIME SERIES

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 5 May 1999	3. REPORT TYPE AND DATE COVERED

4. TITLE AND SUBTITLE Error statistics of time-delay embedding prediction on chaotic time series	5. FUNDING NUMBERS
6. AUTHOR(S) Wood, Joshua T.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Naval Academy Annapolis, MD	8. PERFORMING ORGANIZATION REPORT NUMBER USNA Trident Scholar project report no. 271 (1999)
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Accepted by the U.S. Trident Scholar Committee

12a. DISTRIBUTION/AVAILABILITY STATEMENT This document has been approved for public release; its distribution is UNLIMITED.	12b. DISTRIBUTION CODE

13. ABSTRACT: This project investigates a statistical method for analyzing the error on predictions made through the process of time-delay-embedding of chaotic time series. When viewed as a time-series, chaotic data appears to be unpredictable and random. A chaotic system actually has an orderly representation when viewed in its proper state space (the space consisting of the pertinent variables of the system.) A very remarkable result from the study of chaotic dynamical systems shows that present in almost any single time series is information from all the variables of the state space. The technique of time-delay-embedding provides a method for making predictions on the evolution of this time series. In this method of prediction, one must choose a parameter k, the number of near neighbors in phase space to fit the model to. This project answers the question by describing an algorithm for determining the largest k such that the model adequately fits the data. A prediction is then made from this model along with confidence intervals which measure the reliability of the expected response. While this project involved many different data sets, the purpose was not to analyze these specific data sets, but to develop a general algorithm which could theoretically be used on any chaotic system.

14. SUBJECT TERMS chaos, time-series embedding, prediction, confidence intervals, dynamical systems, noise.	15. NUMBER OF PAGES
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

U.S.N.A. --- Trident Scholar project report; no 271 (1999)

ERROR STATISTICS OF TIME-DELAY EMBEDDING PREDICTION ON CHAOTIC TIME SERIES

by

Midshipman Joshua T. Wood, Class of 1999
United States Naval Academy
Annapolis, Maryland

(signature)

Certification of Adviser Approval

Assistant Professor Erik M. Bollt
Department of Mathematics

(signature)

5/5/99
(date)

Acceptance for the Trident Scholar Committee

Professor Joyce E. Shade
Chair, Trident Scholar Committee

J. E. Shade
(signature)

5 May 1999
(date)

ABSTRACT

This project investigates a statistical method for analyzing the error on predictions made through the process of time-delay-embedding of chaotic time series.

When viewed as a time-series, chaotic data appears to be unpredictable and random. A chaotic system actually has an orderly representation when viewed in its proper state space (the space consisting of the pertinent variables of the system.) A very remarkable result from the study of chaotic dynamical systems shows that present in almost any single time series is information from all the variables of the state space. The technique of time-delay-embedding provides a method for making predictions on the evolution of this time series.

In this method of prediction, one must choose a parameter k, the number of near neighbors in phase space to fit the model to. This project answers the question by describing an algorithm for determining the largest k such that the model adequately fits the data. A prediction is then made from this model along with confidence intervals which measure the reliability of the expected response. While this project involved many different data sets, the purpose was not to analyze these specific data sets, but to develop a general algorithm which could theoretically be used on any chaotic system.

KEYWORDS: chaos, time-series embedding, prediction, confidence intervals, dynamical systems, noise.

ACKNOWLEDGMENTS

This project would not have been possible without help from the following people. I would like to thank Professor Gary Fowler, USNA Mathematics Department, for the help he provided with statistics. I would like to thank Professor Reza Malek-Madani, USNA Mathematics Department, for his help in providing computing equipment, expertise, and guidance. Most of all, I would like to thank my advisor, Prof. Erik Bollt, USNA Mathematics Department, for helping me to see this project to completion.

Contents

1. INTRODUCTION

1.1. A COMPARISON OF TWO TIME SERIES

A great deal of work has been done to study erratic time series. In Figures (1.1) and (1.2), we present two different time series, one from an actual chemical experiment, the other is from a hypothetical experiment.

The time series in Figure (1.1) was generated from the Belousov-Zhabotinski (BZ) reaction in a continuous-flow stirred-tank reactor. The data was collected in a forty hour laboratory experiment by Dr. Milos Dolnik at Brandeis University, Department of Chemistry and Center for Complex Systems. Instead of monotonically approaching an equilibrium state, the concentrations of the reactants oscillate sporadically. The time series in Figure (1.2) was generated from a random source to provide an example of a stochastic time series.

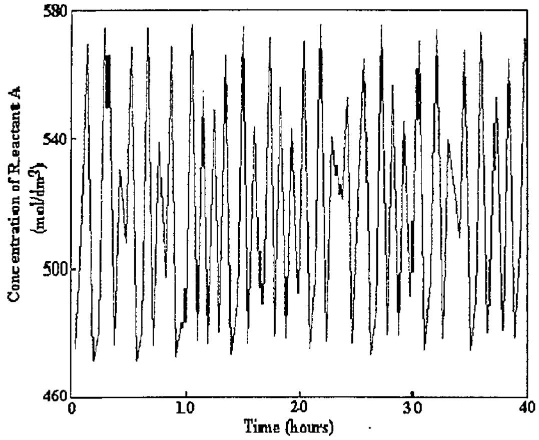

Figure 1.1: Time series from BZ reaction.

Suppose that we suspect the BZ reaction to be deterministic. How could we predict the evolution of the system from a current state? One way would be to model the system based on theoretical considerations of chemistry and records of actual data. The BZ reaction has been modeled with three nonlinear ODE's (Györgyi and Field, 1992). In science such an approach is desirable, but not always possible, especially in a case where all one has is a time series, without knowledge of the system which produced it.

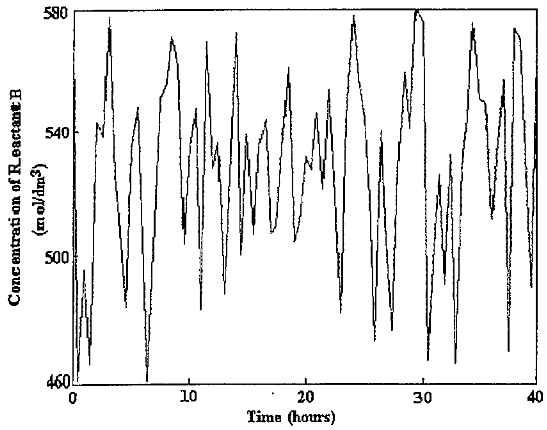

Figure 1.2: Time series generated from a stochastic source.

We will use techniques from recent developments in nonlinear time series analysis to make predictions on the evolution of this time series. These methods provide us a very general approach to this analysis. We will not use theoretical considerations, nor time series from the other reactants, to generate a global model of the system. The method revolves around producing a delay plot, which is a plot of the dependent variable versus itself at previous times.

In Figure (1.3), τ is the time interval between data samples. The fact that this representation of the data appears to follow a curve, leads us to believe that the data is generated from a deterministic process in such a way that each sample is determined by the previous sample: $x_{n+1} = f(x_n)$. Note the difference between the delay plot of the BZ data with the delay plot of the stochastic data in Figure (1.4).

Judging from this representation of the second time series, there is not any connection between one sample and the next. In general, this is how stochastic data will appear. The delay plot will appear as a cloud of data points.

The natural way to make a prediction on the evolution of the first system would be to draw a curve through the data, plug in a value on the x-axis, and read off the value on the y-axis from the curve. It is important to remember that the first time series was not generated to provide an example of a procedure which we could theoretically apply to a physical system; the data was produced from a physical laboratory experiment.

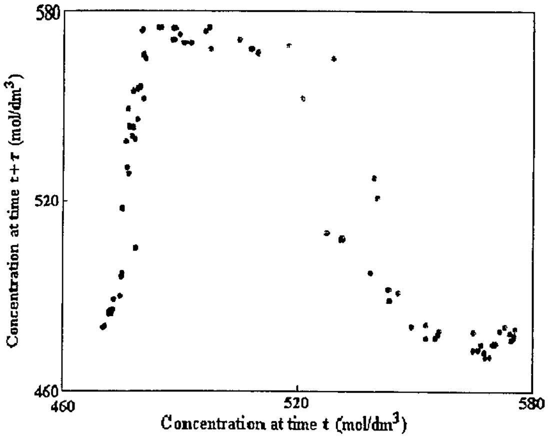

Figure 1.3: BZ reaction data as delay plot.

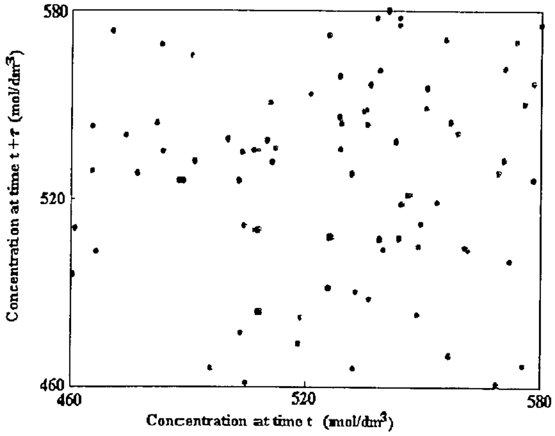

Figure 1.4: Delay plot of stochastic data.

1.2. OUR CONTRIBUTION TO THE PHYSICAL SCIENTIST'S TOOLBOX

Say we are given a time series and are asked to make a prediction on its future evolution. First, we would need to embed the time series in delay coordinates. Determining a good embedding from a time series is not a trivial task. Current literature addresses techniques to select the time delay and the embedding dimension (i.e. number of delay axes) (Kantz and Schreiber, 1997; Abarbanel, 1996). In this project, we assume that we have a good embedding (time delay embedding will be discussed in detail in section 2.5). The state of the art is to linearly regress the k nearest neighbors of the point to be predicted. How best to choose k is not addressed in the literature (Kantz and Schreiber, 1997; Abarbanel, 1996). We would like to bring a statistical approach to this dynamical systems question. The tool we develop will do the following.

1) *We will select the appropriate number, k, of near neighbors such that the model adequately fits the data.* There is a trade-off between choosing a large k vice a small k. The number of near neighbors we choose for prediction will dictate the size of the neighborhood to which we fit our polynomial. Taylor's Theorem tells us to choose a small k to minimize the local truncation error on the polynomial approximation. Because the data contains noise, we would like to use a large k in order to minimize the effect the errors have on the polynomial fit. Using data from a large neighborhood, though requires a higher order polynomial, which then requires more data. We determine the largest k such that the model is appropriate for the data.

2) *We will put confidence bounds on the prediction such that we are able to deal with an interval prediction versus a point prediction.* Current technology allows time series embedding prediction to provide a point prediction, but does not provide a statement about the quality of that prediction (Abarbanel, 1996; Kantz and Schreiber, 1997; Sauer, 1994). By providing a region which we are, say, 95% certain the true value of the prediction lies within, we are providing a means to measure the reliability of the prediction. The model, along with the data to which it is fit, will provide the confidence interval. Since our model is well fit, the confidence interval will be appropriate.

2. CHAOTIC DYNAMICAL SYSTEMS

2.1. DYNAMICAL SYSTEMS

In order to talk about the statistical analysis done on chaotic dynamical systems, we must first describe what we mean by a dynamical system. We begin with an intuitive/physical description, and then give a rigorous mathematical definition.

It is enlightening to compare a dynamical system to an experiment involving a physical system. This system could be a pendulum, a kettle of water, a billiard table, or an internal combustion engine. In this experiment, we are interested in observing how certain quantitative variables evolve over time. For the pendulum, these variables might be the position and velocity of the bob. For the internal combustion engine, these might be oil temperature and pressure, temperature of engine block, and rate of fuel consumption. We start the experiment running and observe what happens to the variables we are interested in. A dynamical system is defined to be deterministic, i.e. there are no random influences on the system. So if we start our experiment in a certain state, collect some data, and then restart it in the original state, and collect a second set of data, the two sets of data will match exactly. Remember that this is not our definition but only an analogy. Now we present the rigorous definition.

We define an n-dimensional dynamical system to be $\Re^n$ (the n-dimensional real numbers) along with a function f (the evolution rule) which describes how any state vector $\mathbf{x} \in \Re^n$ evolves in time. We will define two different types of dynamical systems. A discrete dynamical system is one in which time is considered to have a smallest increment. The evolution rule is a continuous function which maps each state vector to the state vector one time unit later. We will describe the state of the system at time t by a vector $\mathbf{x}_t \in \Re^n$. Let $\mathbf{x}_0 \in \Re^n$ describe the initial state of the system, then for every $t \in \mathbf{N}$ (natural numbers) inductively define $\mathbf{x}_t$ by applying f to $\mathbf{x}_{t-1}$, i.e. $\mathbf{x}_t = f(\mathbf{x}_{t-1})$. We define the trajectory of the dynamical system starting from an initial condition $\mathbf{x}_0$ as the infinite sequence $\{\mathbf{x}_0, \mathbf{x}_1 = f(\mathbf{x}_0), \mathbf{x}_2 = f(\mathbf{x}_1), ...\} = \{\mathbf{x}_t\}_{t=0}^{\infty}$. The trajectory can be thought of as the path through which the dynamical system evolves from the initial condition $\mathbf{x}_0$.

The other type of dynamical system is one in which we consider time to be continuous. The evolution rule for a continuous dynamical system is defined by an ordinary differential equation $\dot{\mathbf{x}} = F(\mathbf{x})$, where $F : \Re^n \rightarrow \Re^n$ is continuous. The state of the system at any time t is denoted as $\mathbf{x}(t)$. Given any initial condition $\mathbf{x}(0) \in \Re^n$, the trajectory of the dynamical system is the solution to the differential equation passing through $\mathbf{x}(0)$.

2.2. CHAOS

Devaney defines a chaotic dynamical system as a system which exhibits the properties of

1) sensitive dependence to initial conditions,
2) topological transitivity, and
3) having topologically dense periodic orbits (Devaney, 1989, p. 50).

These properties are defined in topological terms. Say we have a topological set J and a function $f : J \rightarrow J$. f is said to be sensitively dependent to initial conditions if there exists a $\delta > 0$, such that, for any $x \in J$ and any neighborhood N of x, there exists $y \in N$ and $n \geq 0$ such that $|f^n(x) - f^n(y)| > \delta$. f is said to be topologically transitive if for any open sets U and $V \subset J$ there exists a $k > 0$ such that $f^k(U) \cap V \neq \emptyset$. Finally, f has topologically dense periodic orbits if for any open set $U \subset J$, there exists $x \in U$ and a $k > 0$ such that $f^k(x) = x$ (Devaney, 1989, p. 50).

Intuitively, these definitions say the following. Say, we observe two different trajectories of a chaotic dynamical system, each beginning from initial conditions very close together. Sensitive dependence to initial conditions says that these trajectories will eventually become far apart. A function is topologically transitive on a set if throughout the entire set there exist initial conditions whose trajectories wander throughout the entire set. A function has topologically dense periodic orbits on a set if throughout the set there are initial conditions which eventually evolve back to themselves. These three properties imply that a chaotic dynamical system will be highly unpredictable, it will blend together, and yet there will remain an element of regularity.

Note that Devaney's definition of chaos applies only to the discrete case. Appropriate modifications of the definitions apply to the continuous case. Instead of mapping a point or neighborhood a discrete number of times, one allows a trajectory or set of trajectories to evolve over time.

2.3. THE LOGISTIC MAP

The logistic map is a one-dimensional example of a discrete chaotic system. Let $\mu = [0, 1]$ denote the unit interval. The logistic map is defined by $f : \mu \rightarrow \mu$, such that $f(x) = 4x(1 - x)$. This map has been studied in great depth and is known to be chaotic. Figure (2.1) shows the time series from the logistic map with initial condition $x_0 = 0.8$. As a time series the data looks very erratic. Since the evolution

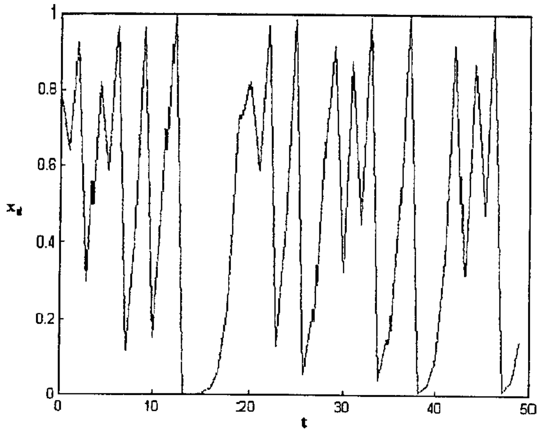

Figure 2.1: Data from the Logistic Map as a time series.

rule maps each x_t to x_{t+1}, the map on the phase space ($\Re$) can be represented in $\Re^2$, as shown in Figure (2.2).

At first sight this example may appear trivial. We generated data using a parabolic map, so of course we can represent the data in an ordered fashion, i.e. along a parabola again. The point is that, though there is a simple map $f : \Re \rightarrow \Re$ which governs the evolution of the system, it is not obvious when the system is viewed as a time series.

2.4. THE LORENZ EQUATIONS

Arguably, the most widely known example of chaos comes from the Lorenz equations. In fact, people who may know very little about chaos overall stand a good chance of having heard of this system. This chaotic dynamical system is given by an ordinary differential equation in $\Re^3$. The Poincaré-Bendixson Theorem (Alligood, Sauer, and Yorke, 1996, p. 337) tells us that we cannot have chaos on a flow in $\Re^2$, so three is the smallest dimension on which we can have a continuous chaotic dynamical system. The Lorenz equations are

$$\begin{aligned}\dot{x} &= \sigma(y-x),\\ \dot{y} &= rx-y-xu\\ \dot{u} &= -Bu+xy,\end{aligned}$$

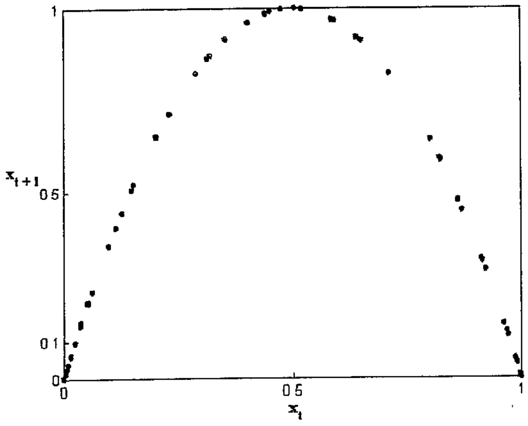

Figure 2.2: Delay plot of Logistic Map data.

where $\sigma = 10$, $r = 28$, and $B = 8/3$ are famous chaotic parameters. We numerically integrate these equations, using the fourth-order Runge-Kutta method (Burden, Faires, and Reynolds, 1978, p. 244), from an initial condition of $\mathbf{x}(0) = (0, 1, 0)$ with a time step $\Delta t = \frac{1}{1000}$ to produce the trajectory in Figure (2.3). This trajectory lies along what is known as the Lorenz attractor. The Lorenz attractor provides a good example of what can happen to a chaotic trajectory. The trajectory does not wind through all of $\Re^3$, but limits on a bounded subset of $\Re^3$.

2.5. TIME SERIES DELAY EMBEDDING

In the introduction we alluded to time delay embedding, now we will discuss it in detail. *Takens's Delay Embedding Theorem* tells us that we can reproduce something equivalent to the phase space attractor of a dynamical system from only one time series of an appropriate observation variable (Takens, 1981). This is truly amazing. It might seem as though we would need to know how all the pertinent variables of a dynamical system evolve in order to characterize the underlying dynamics of the system. This is not the case. One variable captures enough information from the other variables to draw conclusions about the entire system.

Say we have a chaotic dynamical system described by a state vector $\mathbf{x}(t) \in \Re^n$. Let $g : \Re^n \to \Re$, be a scalar observation function which takes any state vector to a corresponding scalar value. This observation function usually projects a trajectory

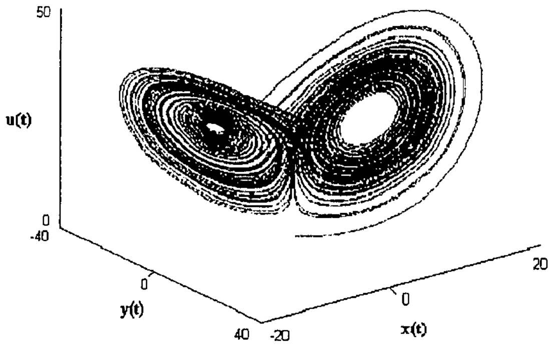

Figure 2.3: The Lorenz attractor, also known as the Lorenz butterfly.

into one of its axes. In other words the function g can "pick off" one of the coordinates of the state vector. For a given embedding dimension M and a time delay τ, we define the delay vector at time t to be

$$\mathbf{Y}(t) = (g(\mathbf{x}(t)), g(\mathbf{x}(t-\tau)), g(\mathbf{x}(t-2\tau)), ..., g(\mathbf{x}(t-(M-1)\tau))).$$

So, $\mathbf{Y}(t)$ is a vector in $\Re^M$ consisting of M observations of the function g taken at equal time intervals. Much work has been done to determine the best method for finding an appropriate embedding dimension and time delay. It has been shown that for an $M \geq 2n+1$ (where n is the dimension of the system), and almost any delay τ, the rule which describes the evolution of $\mathbf{Y}(t)$ is topologically equivalent to the one for $\mathbf{x}(t)$.

2.6. THE LORENZ DELAY PLOT

The delay plot for the logistic map is perhaps not enlightening or surprising. This is because for the natural choice of $\tau = 1$ time unit, the delay plot is identical to a plot of the map of the evolution rule in phase space.

The delay plot for the Lorenz equations is very interesting. Recall that for the Lorenz equations $n = 3$. We would therefore be guaranteed that a delay dimension of $M \geq 2n+1 = 7$ will work. In this case, it turns out that $M = 3$ is sufficient. We denote the state of the Lorenz system at time t by

$$\mathbf{x}(t) = (x(t), y(t), u(t)).$$

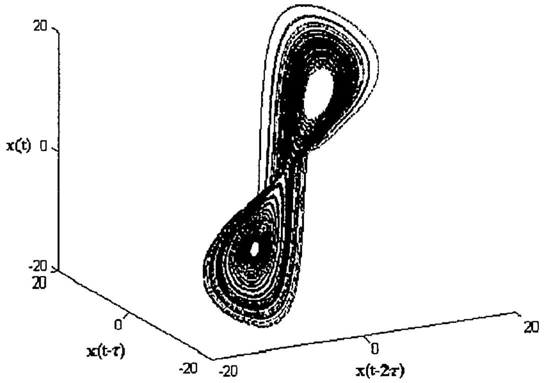

Figure 2.4: Topological reconstruction of the Lorenz attractor generated from only the x time series.

To make observations on the first coordinate, we set $g(\mathbf{x}(t)) = x(t)$. $\tau = 0.05$ yields the delay vector

$$\mathbf{Y}(t) = (x(t), x(t-0.05), x(t-0.1)).$$

Figure (2.4) shows a plot of all the delay vectors. Note that the attractor in the time delay coordinates looks very similar to the Lorenz attractor.

Bear in mind that the delay plot is generated solely from the apparently sporadic $x(t)$ time series in Figure (2.5). We need not consider the $y(t)$ and $u(t)$ time series, nor even be aware of their existences, to make the delay portrait.

2.7. TIME DELAY EMBEDDING PREDICTION

Time delay embedding is not only used to characterize the complete dynamical system in phase space, it is also used to make predictions on an observed time series. Say we have time series $\{x(t_i)\}_{i=0}^{a}$. We will represent this time series as a discrete sequence of points. This is reasonable even if the signal producing the time series is a continuous one, since in an actual experiment we will only be able to sample the signal a finite number of times over some finite time interval.

We then select an M and a τ and embed the time series by considering each delay vector:

$$\mathbf{Y}(t_j) = (x(t_j), x(t_j - \tau), x(t_j - 2\tau), ..., x(t_j - (M-1)\tau)).$$

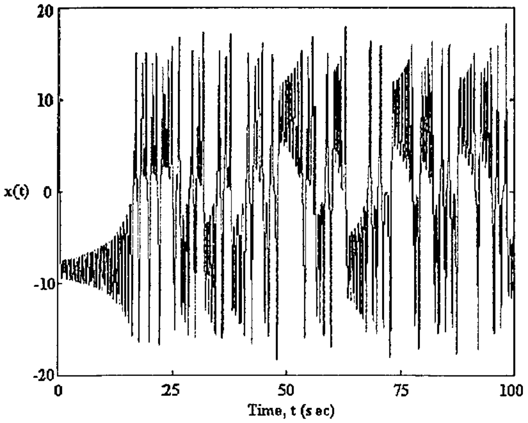

Figure 2.5: Chaotic time series from the Lorenz system used to generate the delay plot.

Say we want to predict the next value of the time series $x(t_{a+1})$, (since $x(t_a)$ is the last point in the time series.) Then in the embedding space $\Re^M$ we locate the k nearest neighbors to $\mathbf{Y}(t_a)$. Denote the evolution rule $\mathbf{G} : \Re^M \rightarrow \Re^M$, such that $\mathbf{G}(\mathbf{Y}(t_j)) = \mathbf{Y}(t_{j+1})$. Since we know where each of the k nearest neighbors is mapped, we can locally approximate $\mathbf{G}$ by linearly regressing these k nearest neighbors. This local approximation of $\mathbf{G}$ is given by $\mathbf{Y}(t_{a+1}) = \mathbf{G}(\mathbf{Y}(t_a)) \cong D\mathbf{G} \cdot (\mathbf{Y}(t_a)) + \mathbf{b}$, where $D\mathbf{G}$ is the Jacobian derivative approximated by regression on the k near neighbors.

This is the general method currently found in the literature (Kantz and Schreiber, 1997; Abarbanel, 1996; Sauer, 1994). Our contribution will be both to decide when the local linear model is statistically significant, or a higher order polynomial is needed, and to say how confident we are of the predictions. To do this we need to discuss the statistical methods we will use.

3. STATISTICS

We now make a diversion into statistics to review some of the tools we require in our analysis of chaotic dynamical systems. We drew heavily from Neter, Wasserman, and Kutner's *Applied Linear Statistical Models*, and Walpole and Myers's *Probability and Statistics for Engineers.*

3.1. CONFIDENCE INTERVALS

In making a prediction we are making an estimation of a quantity. We would eventually like to say how good the estimation is. Let us get away from predictions for a while and begin with a simpler idea: the mean of a set of data. Suppose we were given a sample set of data and asked to estimate the population mean from the data. We could sum the data and then divide by the number of data points. Would this sample mean provide a good estimate of the population mean? It certainly provides the best point estimate of the population mean, but that does not provide an answer to the question of quality. Statistics answers the question by providing a *range* of estimation instead of just a *point* estimate. This range is known as a *confidence interval.*

Suppose that we are trying to estimate a statistical property of a population like the mean and that for any given sample of the population, we developed an algorithm which defined a range of values. This range of values either includes the true population mean or it does not. Say we have multiple samples of the population, each yielding a range of values by the same algorithm. If we know the population value of the statistic we are attempting to estimate, we can determine the percentage of intervals which contain the value of the population statistic. Suppose this percentage is 90%. Then given a random sample of data from the population, we will be able to determine a range of values which have a 90% chance of containing the population statistic. We could then call this range of values a 90% confidence interval.

Now, the hypotheses of the above situation will not apply for most experimentally obtained data sets. For, if we know the true value of the population statistic which we are attempting to estimate, we would not need to estimate it. This idea of a confidence interval is very useful if we know the distribution of the population we are sampling. But, if all we have access to is raw data, how can we know the distribution of the population. For the mean of the population, this question is answered by the Central Limit Theorem.

3.2. THE CENTRAL LIMIT THEOREM

Suppose that we are taking independent random samples from a population. Say we take a number of different samples, each containing n data points. The Central Limit Theorem tells us that as n approaches infinity, the distribution of sample means will limit on a normal distribution, the graph of which is the famous bell shaped curve (we will rigorously examine the normal distribution in the next section).

Figure (3.1) illustrates this fact. Various random samples of data were taken

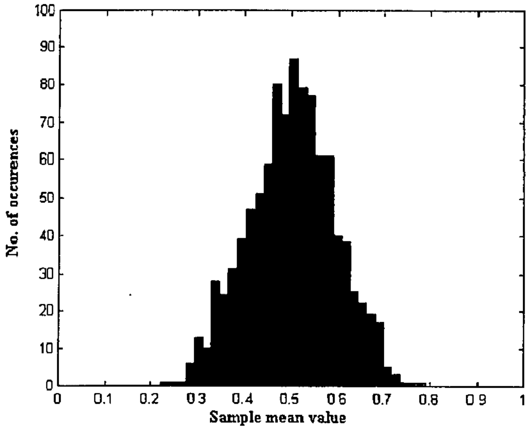

Figure 3.1: Histogram of sample means taken from a population with a uniform distribution.

from the uniform distribution. Each sample of data contained ten points. One-thousand samples were taken, and then these sample means were plotted as a histogram. One should note that the histogram is roughly bell-shaped.

3.3. THE NORMAL DISTRIBUTION

For a continuous random variable, the probability density function (PDF) is defined as a function such that the area under the function between two values is the probability that a random sample of the variable will lie between the two values. For the normal distribution, the PDF is a bell-shaped curve. The normal distribution has two parameters which completely determine its shape. These parameters are the mean (μ) and the standard deviation (σ). The mean locates the center of the curve, and the standard deviation determines how flat or sharp the curve is. The equation for the normal PDF is

$$n(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(1/2)[(x-\mu)/\sigma]^2}.$$

Therefore the probability that a random sample of X will lie between x_1 and x_2 is given by the following equation,

$$\Pr(x_1 < X < x_2) = \frac{1}{\sqrt{2\pi}\sigma} \int_{x_1}^{x_2} e^{-(1/2)[(x-\mu)/\sigma]^2} dx.$$

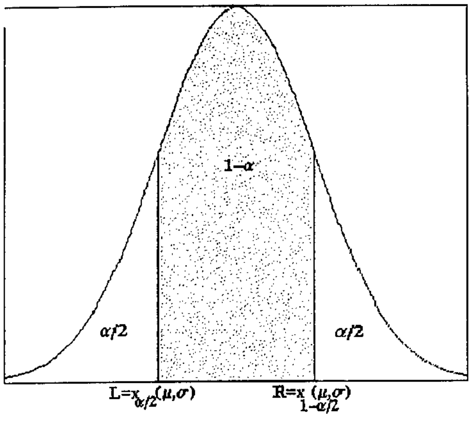

Figure 3.2: Confidence interval for normal distribution. Confidence level is $1 - \alpha$.

For a given μ, σ, and α, define $x_\alpha(\mu, \sigma)$ to be the value of x which satisfies the following equation,

$$\alpha = \int_{-\infty}^{x_\alpha(\mu,\sigma)} \frac{1}{\sqrt{2\pi}\sigma} e^{-(1/2)[(x-\mu)/\sigma]^2} dx.$$

In our applications we use a computer program (Jones, 1993) to determine the value of $x_\alpha(\mu, \sigma)$, for a given α, μ, and σ.

We now know enough to determine confidence intervals on a normal random variable X. Say we know the values of μ and σ, and we want to determine a $1 - \alpha$ confidence interval centered at $x = \mu$. We divide the area under the curve into three sections. In the middle section we have $1 - \alpha$ of the area, which leaves $\alpha/2$ of the area on the left and right (since the total area under the PDF is 1.) The left and right endpoints of the interval are then $L = x_{\alpha/2}(\mu, \sigma)$ and $R = x_{1-\alpha/2}(\mu, \sigma)$ as shown in Figure (3.2) (Walpole and Myers, 1972).

3.4. LINEAR REGRESSION

Now instead of having a problem with one random variable, let us suppose that we have two variables, which depend on each other. By plotting the data on a graph, we could get a rough idea of whether or not the two variables have a linear dependence. Suppose that we have the capability to set a variable X to several different values and then we can measure the variable Y for each value of X. We then plot the data and, if the data lies approximately along a line, we would like to determine the line of best fit. The statistical technique for determining this line is known as linear least squares regression.

We assume that we can set the independent variable X with a high degree of accuracy. We then observe Y and assume that Y contains some random error. We denote the ith observation as (x_i, y_i). We will assume that there is a linear relationship between Y and X, but because of some random error the observations do not lie exactly on a line. We denote this relationship by the following.

$$Y_i = \lambda + \beta x_i + E_i,$$

where E is a Gaussian noise term with mean $\mu = 0$.

3.5. ESTIMATORS OF SLOPE AND INTERCEPT

Given the collection of data $\{(x_i, y_i) : i \in \{1, 2, ..., n\}\}$ for some n, we would like to know the equation for the line of best fit through the data. We will need to determine a, the intercept, and b, the slope of this line. The equation of this line is given by

$$\acute{y} = a + bx,$$

where $\acute{y}$ is the predicted value of y for a given x.

The best line is defined to be the line that minimizes the sum of the squares of the distance that each y_i lies from this line. We therefore want to minimize

$$SSE = \sum_{i=1}^{n} (y_i - a - bx_i)^2.$$

To do this we set $\frac{\partial(SSE)}{\partial a} = \frac{\partial(SSE)}{\partial b} = 0$ and solve for a and b.

$$\frac{\partial(SSE)}{\partial a} = -2 \sum_{i=1}^{n} (y_i - a - bx_i) = 0.$$

So,

$$\sum_{i=1}^{n} (y_i - a - bx_i) = 0,$$

$$\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} a - \sum_{i=1}^{n} bx_i = 0.$$

Finally,

$$na + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i.$$

Now for the other partial.

$$\frac{\partial(SSE)}{\partial b} = -2\sum_{i=1}^{n}[(y_i - a - bx_i)x_i] = 0,$$

$$\sum_{i=1}^{n}(x_i y_i - a x_i - b x_i^2) = 0.$$

Finally,

$$a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i.$$

These two equations are known as the normal equations for this model. Solving for a and b yields,

$$b = \frac{n * SXY - SX * SY}{n * SX2 - (SX)^2},$$

$$a = \bar{y} - b\bar{x},$$

where, $SXY = \sum_{i=1}^{n} x_i y_i$, $SX2 = \sum_{i=1}^{n} x_i^2$, $SX = \sum_{i=1}^{n} x_i$, $SY = \sum_{i=1}^{n} y_i$,and $\bar{y}$ and $\bar{x}$ are the average values of the observations.

It is important to keep in mind that Y is a random variable. Therefore, for different observations of the Y_i's we will get different data sets. Each different data set will yield a different estimate of the true slope and intercept. Some of these estimates will be better than others. Therefore, it is useful to think of the regressed slope and intercept as random variables also. Given a set of data it would therefore be useful to know how well the regressed sample values (a, b) approximate the true population values (λ, β). To do this, we must introduce a new distribution.

3.6. THE t-DISTRIBUTION

In order to determine confidence intervals on the estimates of the slope and intercept of the line of best fit we must use the *(Student) t-distribution.* The PDF of the t-distribution is given as follows,

$$h(t) = \frac{\Gamma[(\nu+1)/2]}{\Gamma(\nu/2)\sqrt{\pi\nu}}(1 + \frac{t^2}{\nu})^{-(\nu+1)/2},$$

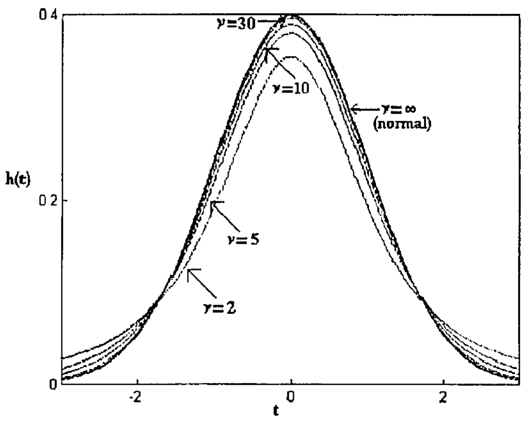

Figure 3.3: The t-distribution for various degrees of freedom, ν. As ν increases the t-distribution limits on the normal distribution.

where ν is the number of degrees of freedom and $\Gamma(z)$ is the gamma function defined as

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx.$$

(Note that $\Gamma(n) = (n-1)!$, for any $n \in \{1, 2, ...\}$.) The t-distribution is used when one does not know the standard deviation of the population that is being sampled and the sample size is too small to assume that the sample standard deviation provides a good estimate of the population standard deviation. The t-distribution, like the normal distribution, is bell-shaped. As the number of degrees of freedom approaches infinity, the t-distribution approaches the normal distribution (see Figure (3.3)).

In a similar manner as we did for the normal distribution, we will define t_α as the value of t such that α of the area lies to the left of t_α. It is important to note that t_α only makes sense if we know the associated degrees of freedom for the distribution (Walpole and Myers, 1972). Again we use a computer algorithm to find its value (Jones, 1993).

3.7. APPROXIMATION OF VARIANCE

We use the t-distribution when we do not know the true value of the population variance σ^2. Instead of σ^2, we use its unbiased estimate MSE (mean squared error).

Say we have a set of data (x_i, y_i). We fit the data with a line $\acute{y} = a + bx$. For each x_i, we determine the predicted response $\acute{y}_i = a + bx_i$. The sum of the squares of the errors SSE is

$$SSE = \sum_{i=1}^{n}(y_i - \acute{y}_i)^2.$$

The mean squared error, defined as

$$MSE = \frac{SSE}{n-p},$$

where p is the number of parameters which were estimated in the regression (in this case $p = 2$), provides an unbiased estimate of the population variance (Neter, Wasserman, and Kutner, 1974, p. 50).

3.8. CONFIDENCE INTERVALS ON THE ESTIMATED SLOPE AND INTERCEPT

Suppose we have a linear model which generates data. Then for each set of n observations, we will be able to generate a confidence interval around our sample a and b. We expect that the appropriate percentage of these intervals will contain the true value of λ and β. The formulas for the $(1-\alpha)$ confidence intervals are as follows.

$$b - \frac{t_{1-\alpha/2}s}{\sqrt{S_{xx}}} < \beta < b + \frac{t_{1-\alpha/2}s}{\sqrt{S_{xx}}}$$

$$a - \frac{t_{1-\alpha/2}s\sqrt{SX2}}{\sqrt{nS_{xx}}} < \lambda < a + \frac{t_{1-\alpha/2}s\sqrt{SX2}}{\sqrt{nS_{xx}}},$$

where s is the estimated standard deviation,

$$s = \sqrt{\frac{SSE}{n-2}},$$

$$S_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2 = SX2 - \frac{SX^2}{n},$$

and the t distribution has $\nu = n - 2$ degrees of freedom.

3.9. MEAN PREDICTED RESPONSE VS. SINGLE PREDICTED RESPONSE

The true response of Y on x_0 is the value of Y the model will return to us if we input x_0. If all we have is a set of data, and we do not know the model from which the data was generated, then we would like to estimate the true response of x_0 with a predicted response in the form of a confidence interval. We can look at a single response or a mean response. A single response is what we expect the model to return from the input of one single x_0. The mean predicted response is what we would expect the model to return on average.

For an estimate a and b, the point predicted response of x_0 is $\acute{y}_0 = a+bx_0$. Of course there is an associated confidence interval for the true response to x_0, denoted y_0.

$$\acute{y}_0 - t_{1-\alpha/2}\sqrt{MSE(1+\frac{1}{n}+\frac{(x_0-\bar{x})^2}{S_{xx}})} < y_0 < \acute{y}_0 + t_{1-\alpha/2}\sqrt{MSE(1+\frac{1}{n}+\frac{(x_0-\bar{x})^2}{S_{xx}})},$$

where again the t-distribution has $\nu = n-2$ degrees of freedom. This formula applies to a single response only. We can also look at confidence intervals on an average response to x_0, whose true value is denoted as $\mu_{Y|x_0}$. The formula for the confidence intervals on the mean response is

$$\acute{y}_0 - t_{1-\alpha/2}\sqrt{MSE(\frac{1}{n}+\frac{(x_0-\bar{x})^2}{S_{xx}})} < \mu_{Y|x_0} < \acute{y}_0 + t_{1-\alpha/2}\sqrt{MSE(\frac{1}{n}+\frac{(x_0-\bar{x})^2}{S_{xx}})}.$$

The confidence interval for the mean response will be narrower than the confidence interval for the single response. This is because the single response has the gaussian error contribution, while in the mean response the gaussian error averages out to zero.

3.10. THE NORMAL EQUATIONS IN MATRIX FORM

We will now demonstrate a more elegant way of looking at least-squares regression, which allows for extension to the higher order problems we consider later. Again, let us suppose that we have a collection of data $\{(x_i, y_i) : i \in \{1, 2, ..., n\}\}$ for some n. Define $\mathbf{Y}$ to be an $n \times 1$ column vector consisting of the y_i's. Define $\mathbf{X}$ to be an $n \times 2$ matrix whose first column is all ones and whose second column is the x_i's. Finally define

$$\mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix},$$

where b_0 is the intercept and b_1 is the slope of the line of best fit to the data.

Let us examine the matrix equation

$$\mathbf{X}'\mathbf{X}\mathbf{b} = \mathbf{X}'\mathbf{Y},$$

where $\mathbf{X}'$ represents the transpose of the matrix $\mathbf{X}$. Our notation to look at the *ith* row and *jth* column of a matrix A will be $(A)_{ij}$. $\mathbf{X}'\mathbf{X}$ is a 2×2 matrix, since $\mathbf{X}'$ is $2 \times n$, and $\mathbf{X}$ is $n \times 2$. The first row of $\mathbf{X}'$ is identical to the first column of $\mathbf{X}$, both of which consist of n ones. Therefore the upper left entry of $\mathbf{X}'\mathbf{X}$ is the sum of n ones, or

$$(\mathbf{X}'\mathbf{X})_{11} = \sum_{i=1}^{n} 1 = n.$$

The upper right entry is the same as the lower left entry, which is the product of a row of ones with a column of x's. So,

$$(\mathbf{X}'\mathbf{X})_{21} = (\mathbf{X}'\mathbf{X})_{12} = \sum_{i=1}^{n}(1)(x_i) = \sum_{i=1}^{n} x_i.$$

Finally, the lower right entry is a product of the row of x's with the column of x's.

$$(\mathbf{X}'\mathbf{X})_{22} = \sum_{i=1}^{n}(x_i)(x_i) = \sum_{i=1}^{n}(x_i)^2.$$

On the right side of the equation we have $\mathbf{X}'\mathbf{Y}$, which is size 2×1.

$$(\mathbf{X}'\mathbf{Y})_{11} = \sum_{i=1}^{n}(1)(y_i) = \sum_{i=1}^{n} y_i$$

$$(\mathbf{X}'\mathbf{Y})_{21} = \sum_{i=1}^{n}(x_i)(y_i) = \sum_{i=1}^{n} x_i y_i$$

Now we see that the equation $\mathbf{X}'\mathbf{X}\mathbf{b} = \mathbf{X}'\mathbf{Y}$ expands to yield the following two equations:

$$nb_0 + b_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$$

$$b_0 \sum_{i=1}^{n} x_i + b_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$$

These equations are the normal equations introduced previously (with a minor renaming of the variables). The least squares estimates for the slope and the intercept, is

$$\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y},$$

assuming that $(\mathbf{X}'\mathbf{X})^{-1}$ exists (Neter, Wasserman, and Kutner, 1974).

3.11. MULTIPLE INDEPENDENT VARIABLES

In our discussion of linear regression thus far, we have assumed that our data was being generated by the model $Y_i = \beta_0 + \beta_1 x_i + E_i$ (again with a slight change of notation). We can also look at models of more than one independent variable. We will denote a model of this type by $Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_k x_{ik} + E_i$, where k represents the number of independent variables which determine Y. Again we denote the least squares estimate to each β_j as b_j. To determine each b_j, we form the sum of squares of the errors, differentiate with respect to each b_j, and set each derivative equal to zero. This yields $k+1$ equations in $k+1$ variables.

$$SSE = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_{i1} - ... - b_k x_{ik})^2$$

For each j,

$$\frac{\partial(SSE)}{\partial b_j} = -2 \sum_{i=1}^{n} [(y_i - b_0 - b_1 x_{i1} - ... - b_k x_{ik}) x_{ij}] = 0$$

realizing that x_{i0} is defined to be identically 1.

So for each j,

$$b_0 \sum_{i=1}^{n} x_{ij} + b_1 \sum_{i=1}^{n} x_{i1} x_{ij} + ... + b_j \sum_{i=1}^{n} (x_{ij})^2 + ... + b_k \sum_{i=1}^{n} x_{ik} x_{ij} = \sum_{i=1}^{n} y_i x_{ij}.$$

Now let us redefine some matrices. For each $i \in \{1, 2, ..., n\}$ and $j \in \{0, 1, 2, ..., k\}$, define $(\mathbf{X})_{ij} = x_{ij}$, the ith observation of the jth independent variable, remembering that $x_{i0} = 1$.

Let

$$\mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ ... \\ b_k \end{bmatrix}.$$

With a little work, one can see that the above system of equations can again be represented by the single matrix equation $\mathbf{X'Xb} = \mathbf{X'Y}$. So again $\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'Y}$.

3.12. ANALYSIS OF VARIANCE (ANOVA)

Recall that each b_i is itself a random variable. It will be useful to compute the variance of each.

Again assume that a true model $Y_i = \beta_0 + \beta_1 x_i + E_i$ is generating data. Assume that each observation E_i is independent and has a variance of σ^2 (denoted $\sigma^2 = Var(E_i)$). Then $Var(Y_i) = \sigma^2$, for each i. Now we compute $Var(b_0)$ and $Var(b_1)$. In order to do this we need the following theorem concerning the variance of the sum of random variables.

THEOREM:

Suppose that $\{Y_i \mid i \in \{1, 2, ..., n\}\}$ is a set of independent random variables. Let $\{a_i \mid i \in \{1, 2, ..., n\}\}$ be a set of constant coefficients. Then

$$Var(a_1 Y_1 + ... + a_i Y_i + ... a_n Y_n) = a_1^2 Var(Y_1) + ... + a_i^2 Var(Y_i) + ... + a_n^2 Var(Y_n).$$

So now for $Var(b_1)$. Recall that

$$b_1 = \frac{n \sum_{i=1}^n x_i Y_i - \sum_{i=1}^n x_i \sum_{i=1}^n Y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}$$
$$= \frac{\sum_{i=1}^n x_i Y_i - \frac{1}{n} \sum_{i=1}^n x_i \sum_{i=1}^n Y_i}{\sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2}.$$

Since,

$$\sum_{i=1}^n (x_i - \bar{x}) Y_i = \sum_{i=1}^n (x_i Y_i - \bar{x} Y_i) = \sum_{i=1}^n x_i Y_i - \bar{x} \sum_{i=1}^n Y_i$$
$$= \sum_{i=1}^n x_i Y_i - \frac{1}{n} \sum_{i=1}^n x_i \sum_{i=1}^n Y_i,$$

and,

$$\sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n (x_i^2 - 2\bar{x} x_i + \bar{x}^2) = \sum_{i=1}^n x_i^2 - 2\bar{x} \sum_{i=1}^n x_i + \sum_{i=1}^n \bar{x}^2$$
$$= \sum_{i=1}^n x_i^2 - \frac{2}{n} (\sum_{i=1}^n x_i)^2 + n\bar{x}^2$$
$$= \sum_{i=1}^n x_i^2 - \frac{2}{n} (\sum_{i=1}^n x_i)^2 + n \frac{(\sum_{i=1}^n x_i)^2}{n^2} = \sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2,$$

we can write b_1 as

$$b_1 = \frac{\sum_{i=1}^n (x_i - \bar{x}) Y_i}{\sum_{i=1}^n (x_i - \bar{x})^2}.$$

By representing b_1 as a sum of the Y_i's,

$$b_1 = \frac{(x_1 - \bar{x})Y_1}{\sum_{i=1}^{n}(x_i - \bar{x})^2} + ... + \frac{(x_i - \bar{x})Y_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2} + ... + \frac{(x_n - \bar{x})Y_n}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

and by the above theorem we derive that

$$Var(b_1)$$

$$= \frac{(x_1 - \bar{x})^2}{(\sum_{i=1}^{n}(x_i - \bar{x})^2)^2} Var(Y_1) + ... + \frac{(x_i - \bar{x})^2}{(\sum_{i=1}^{n}(x_i - \bar{x})^2)^2} Var(Y_i) + ...$$

$$+ \frac{(x_n - \bar{x})^2}{(\sum_{i=1}^{n}(x_i - \bar{x})^2)^2} Var(Y_n) = \sigma^2 \frac{\sum_{i=1}^{n}(x_n - \bar{x})^2}{(\sum_{i=1}^{n}(x_i - \bar{x})^2)^2} = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

Now for b_0. Recall that

$$b_0 = \bar{y} - b_1\bar{x}.$$

So (by the above theorem)

$$Var(b_0) = Var(\bar{y}) + \bar{x}^2 Var(b_1).$$

Now we compute,

$$Var(\bar{y}) = Var(\frac{1}{n}\sum_{i=1}^{n} Y_i) = Var(\frac{1}{n}Y_1 + ... + \frac{1}{n}Y_i + ... + \frac{1}{n}Y_n)$$

$$= \frac{1}{n^2}Var(Y_1) + ... + \frac{1}{n^2}Var(Y_i) + ... + \frac{1}{n^2}Var(Y_n)$$

$$= \frac{\sigma^2}{n^2} + ... + \frac{\sigma^2}{n^2} = \frac{n\sigma^2}{n^2} = \frac{1}{n}\sigma^2.$$

And,

$$\bar{x}^2 Var(b_1) = \frac{1}{n^2}(\sum_{i=1}^{n} x_i)^2 \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sigma^2}{n^2}\frac{(\sum_{i=1}^{n} x_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

So,

$$Var(b_0) = \frac{1}{n}\sigma^2 + \frac{\sigma^2}{n^2}\frac{(\sum_{i=1}^{n} x_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \sigma^2 \frac{n\sum_{i=1}^{n}(x_i - \bar{x})^2 + (\sum_{i=1}^{n} x_i)^2}{n^2 \sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \sigma^2 \frac{n\sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2 + (\sum_{i=1}^n x_i)^2}{n^2 \sum_{i=1}^n (x_i - \bar{x})^2} = \sigma^2 \frac{\sum_{i=1}^n x_i^2}{n\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$= \sigma^2 \frac{\sum_{i=1}^n x_i^2}{n\sum_{i=1}^n {x_i}^2 - (\sum_{i=1}^n x_i)^2}.$$

So,

$$Var(b_1) = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2},$$

and,

$$Var(b_0) = \sigma^2 \frac{\sum_{i=1}^n x_i^2}{n\sum_{i=1}^n {x_i}^2 - (\sum_{i=1}^n x_i)^2}.$$

After that harrowing derivation we will again demonstrate the conciseness of using matrix notation. We define the variance of a $n \times 1$ column vector $\mathbf{d}$ to be an $n \times n$ matrix where

$$(Var(\mathbf{d}))_{ij} = Cov(d_i, d_j).$$

Even though we have not defined *cov* (the covariance of two random variables), it is sufficient to know that $Cov(d_i, d_i) = Var(d_i)$. It is true that $Var(\mathbf{b}) = \boldsymbol{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}$. We will now show that $(Var(\mathbf{b}))_{11} = Var(b_0)$ and that $(Var(\mathbf{b}))_{22} = Var(b_1)$.

$$\mathbf{X}'\mathbf{X} = \begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix}.$$

So

$$\sigma^2(\mathbf{X}'\mathbf{X})^{-1} = \frac{\sigma^2}{n\sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \begin{bmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{bmatrix}.$$

Indeed our assertion that $(Var(\mathbf{b}))_{11} = Var(b_0)$ and $(Var(\mathbf{b}))_{22} = Var(b_1)$ is true (Neter, Wasserman, and Kutner, 1974).

3.13. CONFIDENCE INTERVALS ON PREDICTIONS

Matrix notation gives us a very concise way of determining confidence intervals on fits of data. Say we want to determine the confidence interval on a response to x_0. First we define,

$$\mathbf{x}_h = \begin{bmatrix} 1 \\ x_0 \end{bmatrix}.$$

Note that the point response is given by $y_h = \acute{y}_0 = \mathbf{x}_h'\mathbf{b}$. We will show that the width of our confidence interval is

$$w = t_{1-\alpha/2}\sqrt{MSE(\mathbf{x}_h'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_h)} = t_{1-\alpha/2}\sqrt{MSE(\frac{1}{n} + \frac{(x_0-\bar{x})^2}{S_{xx}})},$$

by showing that

$$\mathbf{x}_h'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_h = \frac{1}{n} + \frac{(x_0-\bar{x})^2}{S_{xx}}.$$

First, expand the matrix notation:

$$\mathbf{x}_h'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_h = \frac{1}{n\sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}[1\ x_0]\begin{bmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{bmatrix}\begin{bmatrix} 1 \\ x_0 \end{bmatrix}.$$

Let us focus on the numerator,

$$\begin{aligned}
[1\ x_0]\begin{bmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{bmatrix}\begin{bmatrix} 1 \\ x_0 \end{bmatrix} &= \sum_{i=1}^n x_i^2 - 2x_0\sum_{i=1}^n x_i + nx_0^2 \\
&= \sum_{i=1}^n x_i^2 - 2\frac{(\sum_{i=1}^n x_i)^2}{n} + 2\frac{(\sum_{i=1}^n x_i)^2}{n} + nx_0^2 - 2x_0\sum_{i=1}^n x_i \\
&= \sum_{i=1}^n x_i^2 - 2\frac{(\sum_{i=1}^n x_i)^2}{n} + 2n\frac{(\sum_{i=1}^n x_i)^2}{n^2} + nx_0^2 - 2x_0\sum_{i=1}^n x_i \\
&= \sum_{i=1}^n x_i^2 - 2\frac{(\sum_{i=1}^n x_i)^2}{n} + 2n\bar{x}^2 + nx_0^2 - 2nx_0\frac{\sum_{i=1}^n x_i}{n} \\
&= \sum_{i=1}^n x_i^2 - 2\bar{x}\sum_{i=1}^n x_i + n\bar{x}^2 + nx_0^2 - 2nx_0\bar{x} + n\bar{x}^2 \\
&= \sum_{i=1}^n (x_i-\bar{x})^2 + n(x_0-\bar{x})^2.
\end{aligned}$$

So the entire fraction equals

$$\begin{aligned}
\frac{\sum_{i=1}^n (x_i-\bar{x})^2 + n(x_0-\bar{x})^2}{n\sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} &= \frac{\sum_{i=1}^n (x_i-\bar{x})^2 + n(x_0-\bar{x})^2}{n\sum_{i=1}^n (x_i-\bar{x})^2} \\
&= \frac{1}{n} + \frac{(x_0-\bar{x})^2}{\sum_{i=1}^n (x_i-\bar{x})^2},
\end{aligned}$$

thus establishing our use of matrix notation. So we now have a concise way of computing confidence intervals on predicted responses.

3.14. HIGHER ORDER POLYNOMIAL FITS TO SINGLE INDEPENDENT VARIABLE

The method of linear regression of multiple independent variables provides an easy way to fit a function of higher powers of a single independent variable to data. We will demonstrate this in the case of a parabolic model. Suppose we have data generated from the model $Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + E_i$. Then we simply adjoin a third column to $\mathbf{X}$, consisting of the square of each x_i, and proceed to compute $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$.

We now illustrate the fitting of higher order polynomials to data. We use the model $Y_i = 1 + 4x_i + 2x_i^2 + E_i$ to generate the following data.

i	x_i	Y_i
1	1.0	3.23
2	2.5	20.54
3	3.0	16.25
4	5.2	73.54
5	6.7	118.76
6	7.3	139.93
7	8.1	179.06
8	9.0	195.49

So

$$\mathbf{X} = \begin{bmatrix} 1 & 1.0 & 1.0 \\ 1 & 2.5 & 6.25 \\ 1 & 3.0 & 9.00 \\ 1 & 5.2 & 27.04 \\ 1 & 6.7 & 44.89 \\ 1 & 7.3 & 53.29 \\ 1 & 8.1 & 65.61 \\ 1 & 9.0 & 81.00 \end{bmatrix},$$

$$\mathbf{Y} = \begin{bmatrix} 3.23 \\ 20.54 \\ 16.25 \\ 73.54 \\ 118.76 \\ 139.93 \\ 179.06 \\ 195.49 \end{bmatrix},$$

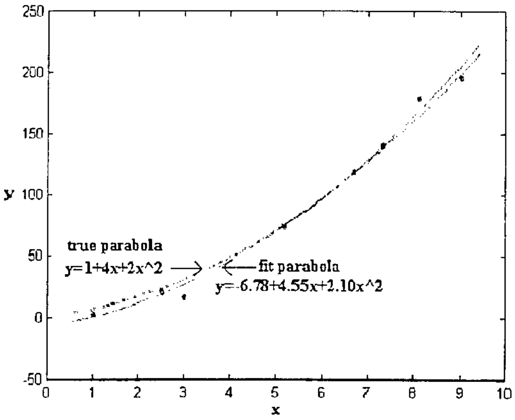

Figure 3.4: True parabola which is used to generate the data versus parabola of best fit.

and we compute

$$\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} = \begin{bmatrix} -6.78 \\ 4.55 \\ 2.10 \end{bmatrix}.$$

Figure (3.4) shows the data, the true parabola, and the fit parabola. By looking at the graph it should appear that the regressed model fits the data better than the true model. The criterion we use to determine how well a model fits data is the sum of the squares of the error (SSE). For the true model, $SSE = 478.1$, while for the regressed model $SSE = 315.18$, the minimum possible value for an SSE obtained from a parabola running through the given data.

Please note that while all the above formulas were proven for either a linear or quadratic fit, the formulas remain the same for a polynomial fit of any degree (Walpole and Myers, p. 335). We simply adjoin columns to $\mathbf{X}$ of the original data raised to successive powers.

3.15. APPROPRIATENESS OF HIGHER ORDER FITS TO DATA

Any analytic function has a Taylor expansion. This means that any well-behaved function can be approximated by a polynomial function. So in terms of regression fitting, we now have the tools to fit a polynomial of any order to data generated from an analytical dynamical system $x_{n+1} = f(x_n)$, in a "small" neighborhood of the point whose evolution we wish to predict.

3.16. STATISTICAL TESTING

Before we jump into the following test, it would be wise to say a few words about general statistical hypothesis testing. Often a statistical test consists of testing whether to accept or reject a statistical hypothesis. The null hypothesis is the hypothesis which will be accepted unless there is evidence to the contrary, in which case we will reject the null hypothesis and conclude the alternative hypothesis, which is the negation of the null hypothesis. There are two types of errors which may be made in this type of testing, usually denoted a Type I error or a Type II error. A Type I error is made when one rejects a true null hypothesis. A Type II error is made when one accepts a false null hypothesis. The alpha level in this test represents the chance of making a Type I error. There is also a beta level which represents the chance of making a Type II error. In most tests of this type, the alpha level is set, but the beta level is unable to be computed. In determining confidence intervals, we do not want to set the alpha level too low, because it makes the confidence interval too large to be useful. In a test of this sort, if we set the alpha level too low, i.e. we choose to rarely reject the null hypothesis, it will drive the beta level up (Walpole and Myers, 1972).

A useful analogy can be made between this sort of testing and a trial in an American court. In court the null hypothesis would be that the defendant is not guilty, since he is "innocent until proven guilty." The alternative hypothesis is that the defendant is guilty. In the American justice system the alpha level is set to be very small; the idea being that we rarely convict an innocent man. We accept the fact that the beta level may be significantly high, as described by the idea that we would rather set free ten guilty persons than convict one innocent man. A very low alpha level corresponds to a level of proof of "beyond a reasonable doubt", while a higher alpha level would represent "clear and convincing evidence" or higher yet "preponderance of the evidence" which is used in several types of military hearings.

3.17. THE t-TEST FOR DEGREE OF POLYNOMIAL, FULL MODEL VERSUS SUB-MODEL

Say we are given a set of data which appears to have a slight curve to it. It would be nice to know whether or not the best fit parabola yields any more information than the linear fit. Is the "slight curve" statistically significant enough to support a quadratic full model over the simpler linear sub-model. The t-test is just such a test. Let us suppose that we fit the function $Y_i = b_0 + b_1 x_i + b_2 x_i^2$ to a set of data. Let $\beta_2 = E(b_2)$, i.e. β_2 is the expected value of b_2. We will test the following set of hypotheses:

$$\begin{aligned} H_O &: \beta_2 = 0 \\ H_A &: \beta_2 \neq 0 \end{aligned}$$

Where H_O is the null hypothesis and H_A is the alternative hypothesis.

To do this we will compute a test statistic t_s and compare it to $t_{(1-\alpha/2,n-3)} = t_{comp}$. The variable t_s is computed as follows.

$$t_s = \frac{b_2}{s\{b_2\}}$$

where $s\{b_2\} = \sqrt{s^2\{b_2\}}$ is the unbiased estimate of the standard deviation of b_2. $s^2\{b_2\}$ is easy to compute. First we compute $s^2\{\mathbf{b}\} = MSE(\mathbf{X}'\mathbf{X})^{-1}$ and then pick off the appropriate value from the matrix. If $t_s < t_{(1-\alpha/2,n-3)}$, we conclude H_O: $\beta_2 = 0$. On the other hand, if $t_s > t_{(1-\alpha/2,n-3)}$, we conclude H_A: $\beta_2 \neq 0$. (Neter, Wasserman, and Kutner, 1974)

We view this procedure as the test of a full model versus its sub-model. The sub-model is valid when the full model degenerates by having a very small leading coefficient.

4. STATISTICS IN DYNAMICAL SYSTEMS

We assume a low dimensional chaotic dynamical system, subject to additive noise, $x_{n+1} = f(x_n) + E_n$. We use the above ideas to select the statistically significant polynomial model in the neighborhood of a point x_p which we wish to predict the response $f(x_p)$. Taylor's theorem gives us good reason to believe that a polynomial will approximate the function in a small enough neighborhood. This represents a significant improvement in the current state of technology in which researchers select a linear model and haphazardly choose several near neighbors. We also offer a statement on the quality of the prediction in terms of confidence. These advancements in dynamical systems theory are made by slight modifications on the above tools borrowed from statistics theory. What follows can be considered to be new technology we are developing for the physical scientist.

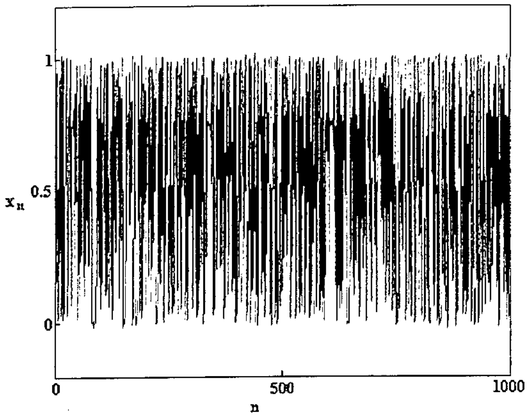

Figure 4.1: Time series from the Logistic Map.

4.1. FINDING K FOR A SPECIFIED DEGREE POLYNOMIAL FIT

We now return to the setting of time delay embedding prediction and ask the question: How do we choose the best number of near neighbors to use in our regression of the data? Say we want to use a linear fit to make predictions on the time series generated from the logistic map with an additive gaussian error term. First, we generate a time series starting from an initial condition of $x_1 = 0.4$ and, for each $n \in \{2, ..., 1000\}$, we set $x_n = f(x_{n-1}) = 4x_{n-1}(1 - x_{n-1})$. We add a normally distributed random noise term to each x_n producing the following time series shown in Figure (4.1).

Say we want to determine what point in the time series would follow $x_0 = 0.35$. First, we embed the time series in one dimension with a time delay of one iterate, and then we regress a line from the k near neighbors to x_0. Now we have to choose a value of k. Let us regress a line from the 300 near neighbors (shown in red in Figure (4.2)) to x_0. It is apparent from Figure (4.2) that the data is significantly curved, and the line of best fit does not accurately fit the data at x_0.

Using the t-test, we compute that the significance level for this fit is $\alpha \approx 0$. This means that we are almost 100% positive that we are not making a Type-I error. Which also means that we are almost 100% positive we are making a Type-II error. Recall that a Type-I error is erroneously calling linear data nonlinear. The way to avoid this type of error 100% of the time is never to call any data nonlinear. To sum

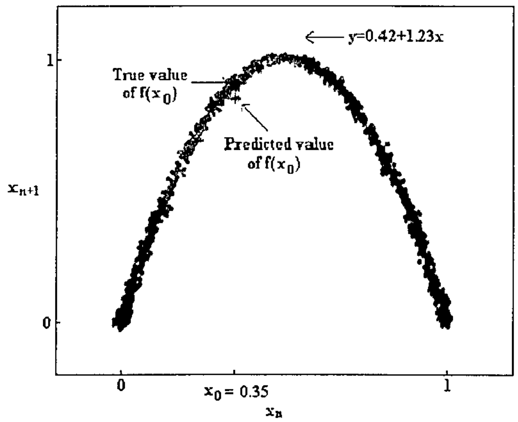

Figure 4.2: Line of best fit, regressed from too many near neighbors.

it up, this is a very bad fit.

Let's examine the flip-side of the same coin. What if we use too few near neighbors? Figure (4.3) shows a fit to the 10 near neighbors of x_0. To help see the problem Figure (4.4) is a close up of the region around $x_0 = 0.35$. Since the slope is clearly wrong, we do not want to use a fit such as this.

What happens in the very small region around x_0 is that the statistical errors become too significant to determine the shape of the attractor. The problem becomes worse if we use a very dense set of points in a very small interval of the x-axis. We have been looking at an attractor with $N = 1000$ points on it. Now lets fill the attractor with $N = 100000$ points and fit a line and a parabola to the 200 near neighbors of $x_0 = 0.35$. Figure (4.5) shows the fitted parabola and line. Figure (4.6) is the close-up view.

What is wrong with these pictures? From the blown up picture, we see that the data appears as a cloud. The error blurs the structure. There is a more significant problem though. We can choose to tighten up the region on the x-axis as much as we like, but we have no control on the y-axis, the standard deviation of the error controls this size. Looking at too small a region on the x-axis produces the phenomenon which we termed the *"tall skinny box"* problem. Note the x-scale versus the y-scale. The increment on the y-axis is about 0.08 units, while the increment on the x-axis is only 0.003 (differing by a factor of approximately 30). Note that both the fits are not very good. The slope of the line is off (imagine a line tangent to

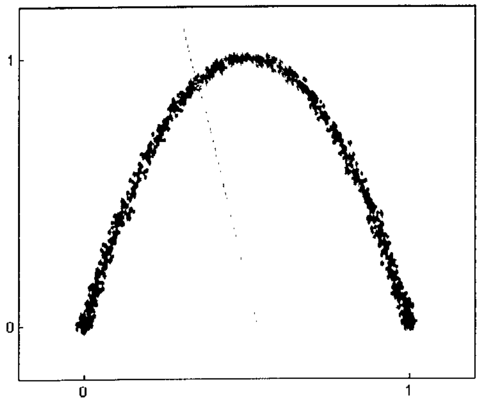

Figure 4.3: Line of best fit, regressed from too few near neighbors.

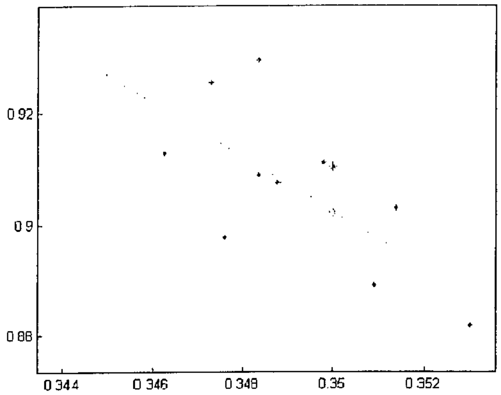

Figure 4.4: Closeup of data used to fit line in previous figure.

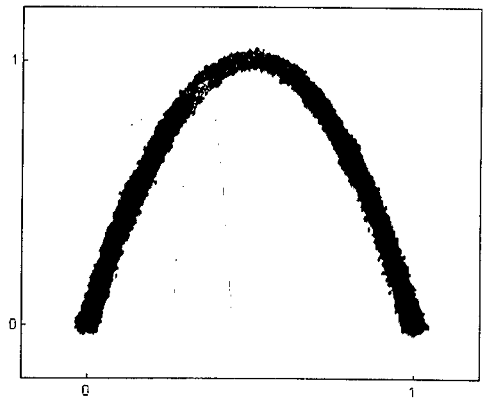

Figure 4.5: Linear and quadratic fits using too few near neighbors on densely filled attractor.

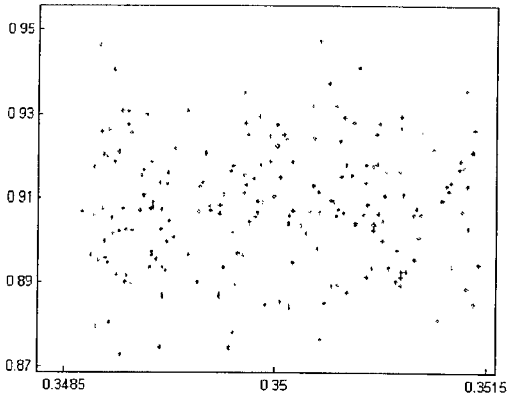

Figure 4.6: Closeup of near neighbors from previous figure.

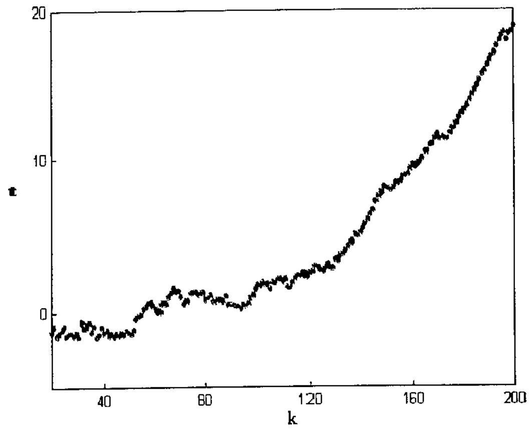

Figure 4.7: Plot demonstrating that the linear model becomes more appropriate as the nearborhood size becomes smaller. $t > 0$ implies that the quadratic map must be used. $t < 0$ implies that the linear map is appropriate.

the attractor) and the parabola is too narrow. In even more extreme cases we get a computer error (much akin to division by zero), since the data begins to line up along a vertical line.

In writing the code for this problem the first step was to start searching for what we call the critical value of k, k_{cr}. We defined k_{cr} as follows. Say we were attempting to fit a line to the k-near neighbors. Then for each value of k we fit the data with a parabola, $y = b_0 + b_1x + b_2x^2$. By doing so we assume that the data is coming from a parabola, $y = \beta_0 + \beta_1x + \beta_2x^2 + E$. We then use the t-test to tell us whether or not $\beta_2 = 0$. I.e. is it really necessary to look for a parabola or would a line fit the data just as well? We call k_{cr} the value of k such that for $k = k_{cr}$, the t-test says use the line, but for $k = k_{cr} + 1$, the t-test says use the parabola.

A problem occurred when using large amounts of data. Look at the logistic map with $N = 1000$, and let's suppose that we want to fit a line through $x_0 = 0.35$. Consider Figure (4.7). On the x-axis is k, the number of near neighbors to x_0. On the y-axis is $t = t_s - t_{comp}$ (see Section 3.17). If $t > 0$, then we conclude that we must use a parabolic fit, but if $t < 0$ a linear fit will suffice.

For this set of data, the way in which we defined k_{cr} is unambiguous. Once the graph crosses into the linear region it stays in the linear region. But for larger amounts of data, k_{cr} may be ambiguously defined i.e. there is not a single crossing

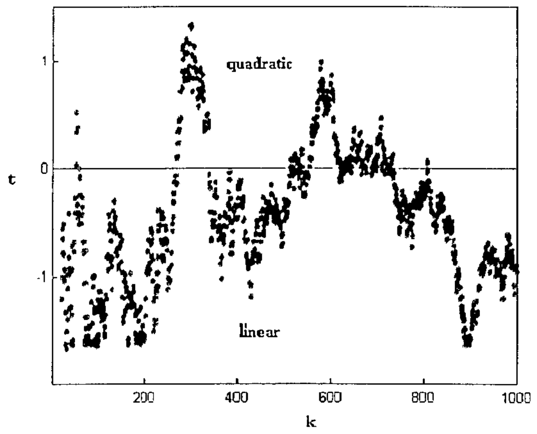

Figure 4.8: Plot of t versus k for very long Logistic time series. The attractor is more densely filled than in the previous figure.

point, but the graph may oscillate between quadratic and linear regions. Figure (4.8) is the plot of t versus k, but this time with $N = 100000$. Notice that by the time we have narrowed our region down to the 1000 nearest neighbors, we are already in the linear region. But around $k = 800$, the graph reenters the quadratic region. It jumps back and forth between the two regions several more times. Therefore, for us to speak of *the* value of k_{cr} is a mistake, as we have defined it, k_{cr} is not unique. Think of it in the following terms. Say we have a region containing k_{cr} near neighbors. The t-test tells us that we have data which appears linear. When we add the next nearest neighbor into the analysis, the model needs the quadratic term (by definition of k_{cr}). It may very well happen that the next nearest neighbor again makes the data appear linear, and will reverse the decision of the t-test back to linear.

The two most logical choices we have for k_{cr} is the largest value or the smallest value of k with the defining property of k_{cr}. Often scientists like to fit data in as small a region as possible. This is because the error term on the Taylor series approximation to the function is governed by the size of the interval. If this were our goal we would use the smallest value of k_{cr}. But as we get to very small values of k we have already seen the tall skinny box problem arise. Also the fluctuations at the extremely small values of k can be thought of as noise, in which case it would be almost meaningless to use any of them. However, this begs another question. What is to say that the initial crossing of the function from the quadratic region to

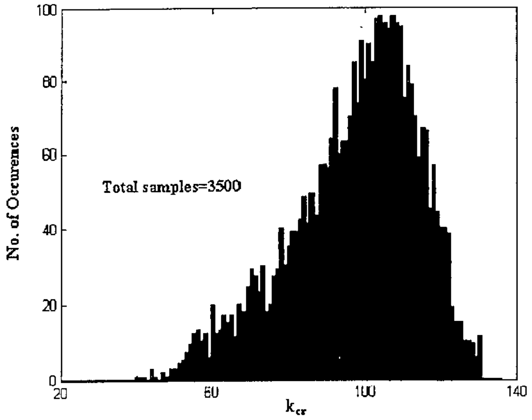

Figure 4.9: Histogram of k_{cr} for the Logistic Map.

the linear region is not itself noise? It certainly varies with the random error, but is it more a function of the attractor or the random error. Figure (4.9) addresses this issue. Since k_{cr} is distributed about some expected value, we conclude that k_{cr} is a function of the attractor and the point to be predicted, x_0. We therefore make the following definition.

Definition: Given a 1-dimensional embedding of a chaotic dynamical system, and an x_p from which to predict $f(x_p)$ based on linear regression of degree m of the k nearest neighbors to x_p, define k_{cr} as follows. $k_{cr} = \max\{k : \beta_{m,k} = 0$ but $\beta_{m,k+1} \neq 0\}$ where $\beta_{m,k}$ is the coefficient on x^m in the regressed model of f about x_p using k near neighbors to x_p.

4.2. BAND SIZE ON RESPONSES

Say we have an embedded time series in 1-dimension and we would like to look at interval responses to $f(x_p)$. Then, for any degree polynomial model, we have determined a way to choose the number of near neighbors to use in the fit of f. Is there an algorithmic way to choose the degree of the polynomial? Let us again look at the data from the BZ reaction. Figure (4.10) shows the confidence bands across the spectrum of $x's$ covered by the embedded data. To make the graph a grid of $x_p's$ was selected. For each x_p, we determined k_{cr} for a linear fit (i.e. the largest number of near neighbors in which the quadratic fit degenerates into a linear fit.)

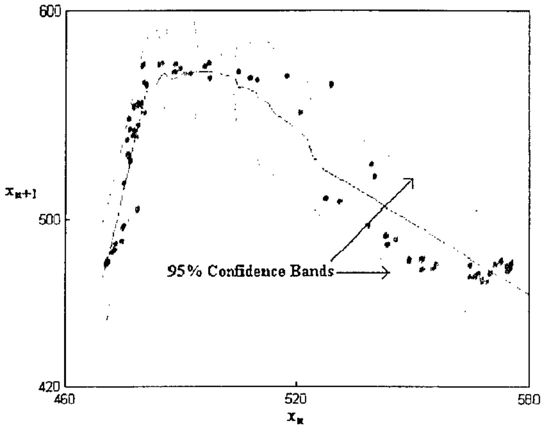

Figure 4.10: Confidence bands from fitting local linear model to BZ reaction data.

This test requires an α level, which is the probability of making a Type-I error. We set $\alpha = 0.1$ which means that for the k_{cr} near neighbors to each x_p, we have a 10% chance of erroneously declaring the linear fit not statistically valid. We then fit the 95% confidence bands to each predicted response.

Figure (4.11) contains a graph which is generated in entirely the same way except that instead of using a linear fit at each x_p, we use a quadratic fit. Note that for most values of x_p the bands from the quadratic fit appear tighter than the linear fit, though there are some places where the linear fit appears smaller. If we need to decide between either the linear or the quadratic in a global sense then one might use the area between the bands as the deciding factor, with the smaller area representing more accurate prediction capability. We ran a rough numerical integration to approximate the area between the bands and found that the area enclosed by the linear fit, $A_1 = 5150$, while the area enclosed by the quadratic fit was $A_2 = 4570$. If we had to settle on one fit we would choose the quadratic, but we do not have to choose one. The predicted responses are not global in nature, they are local to each x_p. Therefore if we are interested in attaining the smallest bands, it might be useful to determine what degree fit to use based on the size of the confidence interval it produces. To produce the global picture this means that at each x_p we would keep the bands which have the smallest width. Figure (4.12) demonstrates what that would look like.

While this procedure is advantageous from the point of view of producing

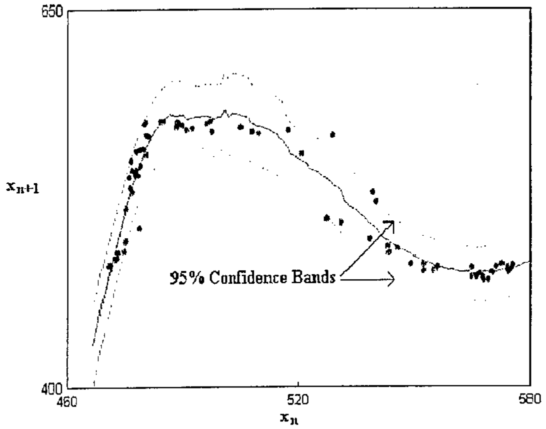

Figure 4.11: Confidence bands from fitting local quadratic model to BZ reaction data.

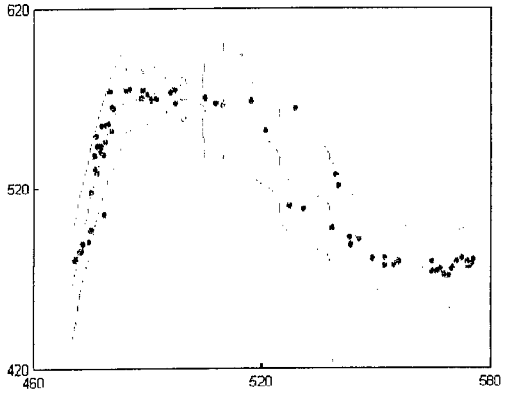

Figure 4.12: Confidence bands formed by pointwise choosing the smaller of the linear or quadratic bands.

smaller bands, it has some drawbacks. The higher order models allow us to use a larger region of data since these models can account for curvature in the data. This increases the number of data points that we are fitting the model to and the number of parameters that we need to fit. Therefore, the algorithm takes up much more time to run on the computer.

Statisticians are often skeptical about fitting higher order polynomials (above cubic) to data. We recommend that for a typical data set the model be restricted to the linear case unless there is some theoretical consideration which suggests that the data tends to fit well by a higher degree polynomial fit.

4.3. BENCHMARKING THE 1-D PROCEDURE

Now we would like to establish that the above described procedure is giving us correct interval predictions. We will look at several different data sets and run slightly different tests on each.

4.3.1. LOGISTIC DATA

The version of the Logistic Map we use to test the algorithm is

$$x_{n+1} = f(x_n) + E_n = 3.85 * x_n(1 - x_n) + E_n.$$

We generate a time series by iterating this map from an initial condition of $x_0 = 0.7$, and at each step we add a normally distributed error with parameters $\mu = 0$ and $\sigma = 0.01$ to each term. We embed the data and from the first 100 values of the time series, we make 90% confidence bands. This is illustrated in the Figure (4.13). In order to test the accuracy of these bands, we then look at the next n embedded points and determine the percentage of points P_n which lie inside the bands. The values for select values of n are given in the chart in Table (4.1).

n	100	1000	10000
P_n	84.0	87.8	87.6

Table 4.1: For a test series n units long, P_n is the percentage of times the true evolution of the system fell within the 90% confidence bands generated from 100 points.

Next we examine what happens as the number of points used to generate the bands is increased. We generate the bands shown in Figure (4.14) by using the first 1000 points of the series. When we test the accuracy of these bands, we get the results shown in Table (4.2).

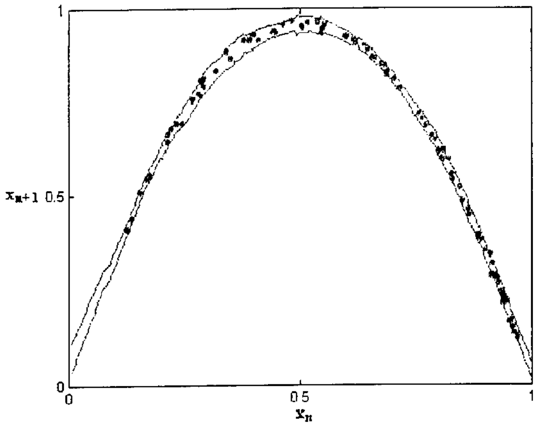

Figure 4.13: The 90% confidence bands generated from a 100 point Logistic Map time series.

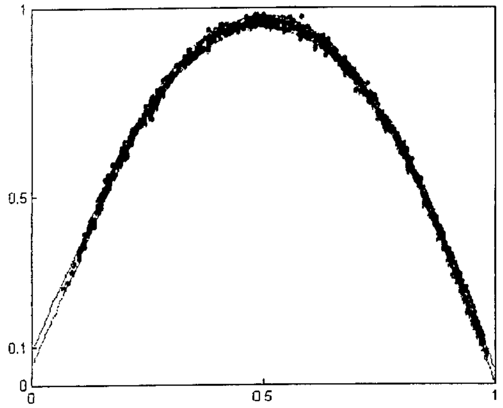

Figure 4.14: The 90% confidence bands generated from a 1000 point Logistic Map time series.

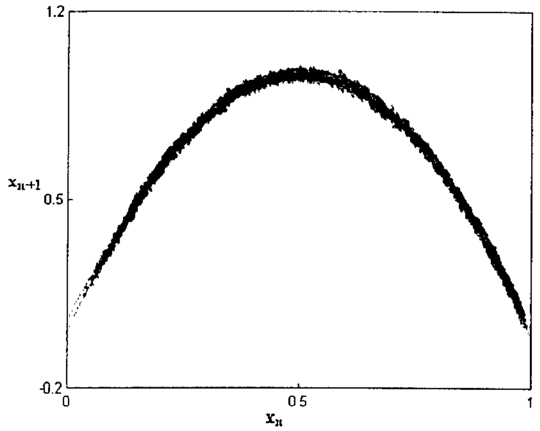

Figure 4.15: The 90% confidence bands generated from a 5000 point Logistic Map time series.

n	100	1000	10000
P_n	92.0	90.6	90.2

Table 4.2: For a test series n units long, P_n is the percentage of times the true evolution of the system fell within the 90% confidence bands generated from 1000 points.

Finally we use 5000 points to generate the 90% confidence bands shown in Figure (4.15). The results are given in Table (4.3).

n	100	1000	10000
P_n	88.0	89.3	90.2

Table 4.3: For a test series n units long, P_n is the percentage of times the true evolution of the system fell within the 90% confidence bands generated from 5000 points.

We should expect the accuracy of the prediction bands to get better as the number of points used to generate them is increased. The data supports this to some extent, though the bands generated from 1000 points appear to do as good a job as the bands generated from 5000 points.

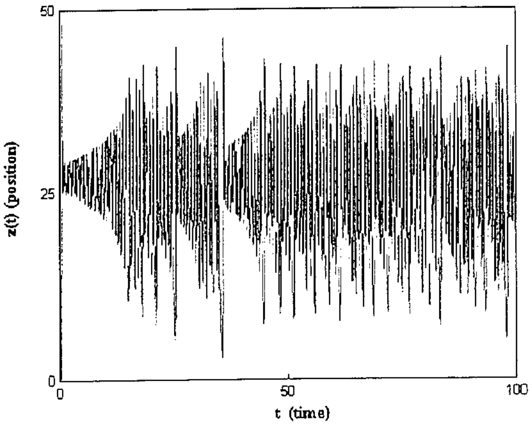

Figure 4.16: Lorenz $z(t)$ time series.

4.3.2. LORENZ INTENSITY RETURN DATA

Consider a Lorenz time series $\{z(t) : t \geq 0\}$. Define t_n as the time at which the nth local maximum occurs. Let $I_n = z(t_n)$. There is a function f, known as an intensity return map, which maps each I_n to $f(I_n) = I_{n+1}$. To see this, we generate the time series $\{z(t) : t \in \{0, ..., 100\}\}$ shown in Figure (4.16). If we plot each local maximum versus its predecessor, we get the cusp in Figure (4.17).

We generate the time series by numerically integrating the Lorenz equations (three ODE's) using the fourth-order Runge-Kutta method (Burden, Faires, and Reynolds, 1978, p. 244). The cusp in Figure (4.17) is generated by choosing a small time step $\Delta t = \frac{1}{1000}$. In order to simulate noise, we increase the time step to $\Delta t = \frac{1}{20}$, and get a much less accurate approximation of the time series. Therefore the intensity return data will contain more error as shown in Figure (4.18).

Now, we generate confidence bands using the first 100 points from the data (Figure (4.19)). For the next n points in the sequence we determine what percentage P_n of the data lies inside the bands and display this in Table (4.4).

n	100	1000	10000	20000	30000
P_n	96.0	96.6	96.0	96.0	95.9

Table 4.4: For a test series n units long, P_n is the percentage of times the true evolution of the system fell within the 90% confidence bands generated from 100 points.

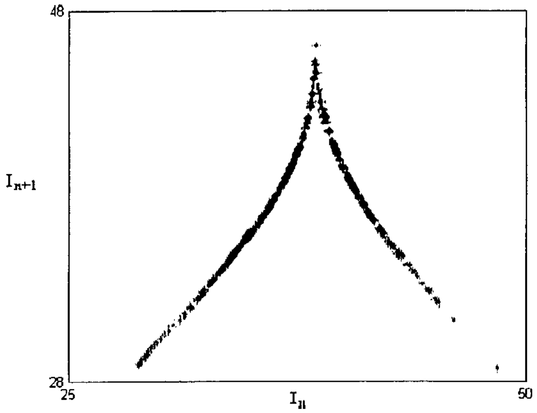

Figure 4.17: Successive maxima plot from z time series of Lorenz system with $\Delta = 0.001$.

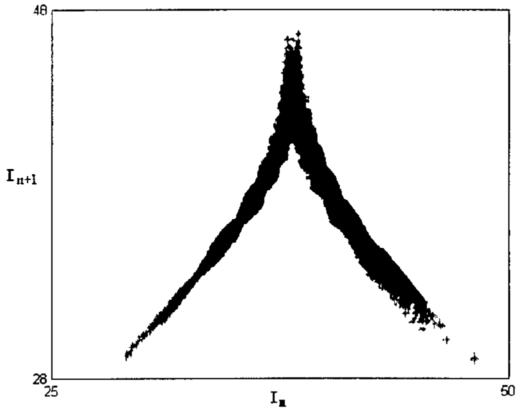

Figure 4.18: Successive maxima plot from z time series of Lorenz system with $\Delta = 0.05$. The large time step introduces a non-gaussian error.

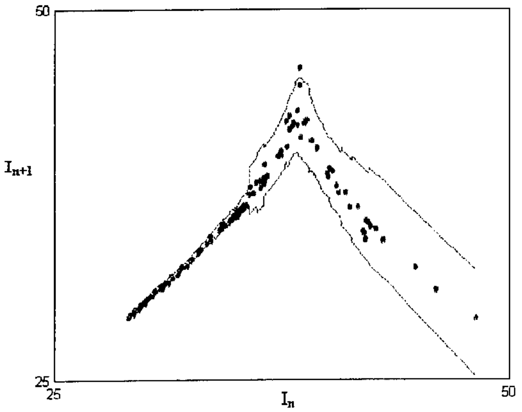

Figure 4.19: The 90% confidence bands generated from the first 100 points of the successive maxima map from the z Lorenz time series.

These bands appear to be 96% confidence bands, instead of the hoped for 90%. Figure (4.20) shows the bands made with 1000 points of the sequence.

n	100	1000	10000	20000	30000
P_n	91.0	91.2	91.5	91.8	91.6

Table 4.5: For a test series n units long, P_n is the percentage of times the true evolution of the system fell within the 90% confidence bands generated from 1000 points.

The results in Table (4.5) show that the bands come reasonably close to capturing 90% of the data.

The statistics involved in these tests assume that the error on the data is from a gaussian distribution. Instead of artificially adding a Gaussian noise term, this method of simulating noise produced an unknown error distribution. This test helps to validate the use of this method to sets on which the distribution is either not known or possibly non-gaussian.

4.3.3. LASER DATA

In 1991, the Santa Fe Institute released several time series for a competition dealing with predicting the evolution of the time series. One of these data sets is a time series of intensity from an NH_3-FIR laser collected by U. Hübner, N. B.

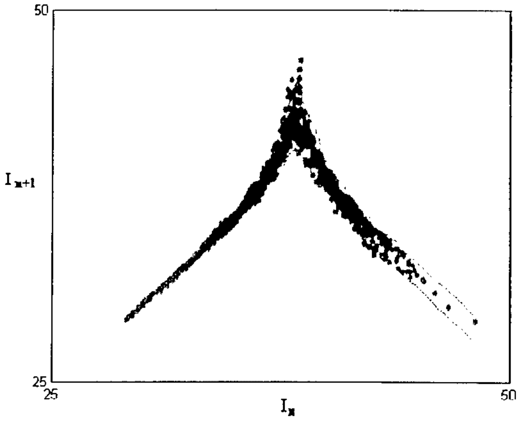

Figure 4.20: The 90% confidence bands generated from the first 1000 points of the successive maxima map from the z Lorenz time series.

Abraham, C. O. Weiss and others at PTB Braunschweig in Germany (Weigend and Gershenfeld, 1993). Figure (4.21) shows the first 1000 points of the series.

Hübner et al. show that the semiclassical laser equations are the same as the Lorenz equations with the parameters chosen correctly. This inspired us to look at the intensity return plot. Figure (4.21) contains only the first 1000 points of the series. The entire series has approximately 10000 data points. In this series there are approximately 1300 local maxima. The intensity return plot and the 90% confidence bands are shown in Figure (4.22).

The center section of the data is well grouped together with only a few outliers. The extremely low values, as well as the extremely high values, of I_n map to a wide range of I_{n+1}'s. The size of the confidence bands reflect the reliability of our predictions. In the regions where our data is dispersed, the confidence bands are wide. Where the data is tightly grouped, the bands are narrow. From the size of the bands, we can separate the range of values of the I_n's into an interval which is "predictable" and two intervals which are "unpredictable."

While our point prediction method does yield a prediction, our interval predictions tell us where we can trust the predictions. One might offer the objection that it is possible to look at the data, see where it is "dispersed", and then decide what predictions are trustworthy and which ones are not. Recall that one of the motivations for this project is to provide an objective method for determining confidence on predictions. What the predictions will be used for will determine how accurate

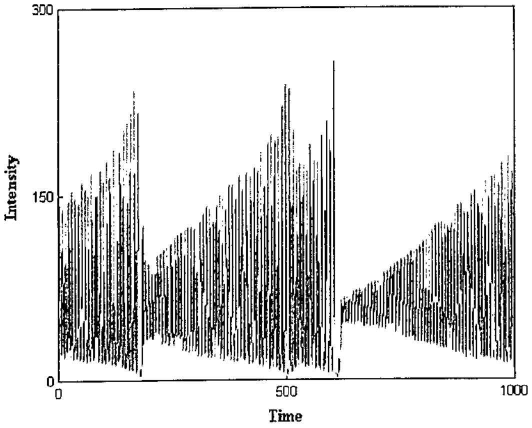

Figure 4.21: NH_3-FIR laser data as time series.

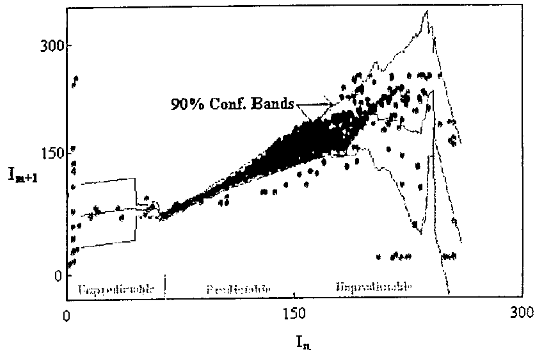

Figure 4.22: Intensity return plot for the NH_3-FIR laser data. This data provides a good example of an embedding which has regions of both high and low predictability.

the predictions need to be. Once a tolerance is decided upon, we can determine (with this method) the width of the confidence interval, and if it exceeds the level of acceptability, then we know not to trust the prediction.

Being able to predict some of the time is better than never being able to predict. *Knowing when we can make a prediction on the data is better than not being able to tell the difference between predictable and unpredictable.* Consider the system which represents the "be all and end all" for time series prediction: the stock market. If an analyst could recognize when the stock market enters a region of predictability, then he could determine when to invest and when to wait. Even though he would not be able to make predictions all of the time, he would be able to wait until the data entered a region of predictability.

The stock market is cursed (from a prediction point of view) with a dimensionality problem. Its dimension is so high that we may not have enough recorded data to be able to adequately cover the attractor. This would be like trying to determine the shape of the Lorenz attractor by covering it with a handful of points. The process of time delay embedding looks back through the record of the time series to find the sections of data which most closely resemble the data near the prediction point. If it takes a large number of delays to be able to uniquely represent a point in the phase space, then a very large time series is necessary to fill out the high-dimensional attractor. This drives up the length of the recorded data set. It may be that the stock market has not been around long enough to see enough of its attractor.

This method lays the ground work for the more general multi-dimensional case. In one dimension, it is possible to look at the data and determine to what degree the data is dispersed. In the multi-dimensional case it is very difficult, if not impossible to visually display the data in an informative way. When this algorithm is generalized to higher dimensions, it will provide a quantitative measure of the reliability of a prediction even when we cannot display the data in 2 or 3 dimensions.

4.4. TIME DELAY EMBEDDING THE LASER DATA

The intensity return map for the laser data does not appear to embed nicely in one dimension. The analysis of confidence bands is useful for demonstrating that the size of the confidence bands in regions of low predictability are large. We will now attempt to time-delay-embed the laser data.

Recall that the time delay vector has the form

$$\mathbf{Y}(t) = (g(\mathbf{x}(t)), g(\mathbf{x}(t-\tau)), g(\mathbf{x}(t-2\tau)), ..., g(\mathbf{x}(t-(M-1)\tau))),$$

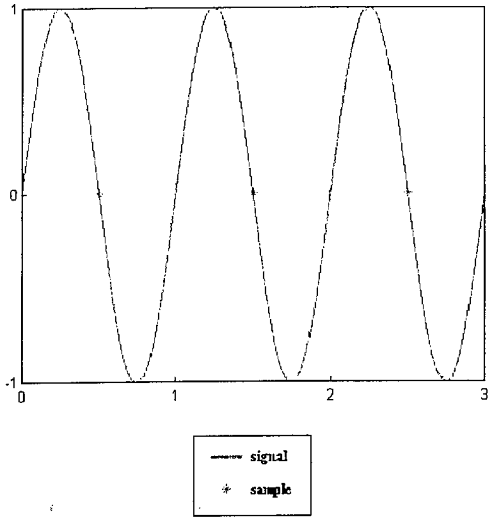

Figure 4.23: Example of a periodic signal being sampled at its natural frequency.

where $\mathbf{x}(t)$ is some unknown vector valued function consisting of all the pertinent variables which drive the output of the laser. Recall that the laser data consists of data observed during an actual experiment. As such, our time series is a discrete sequence, which we will denote as $\{x(t)\}_{t=1}^{N}$, for some large N. Therefore our observation function g must map the state vector to our data points, $g(\mathbf{x}(t)) = x(t)$. We will now illustrate some possible ways to choose τ and M.

4.4.1. CHOOSING τ

Consider the function of $x = sin(2\pi t)$. How many times should we sample this function to get an adequate idea of what the signal looks like? The period of this function is $T = 1$. If we sample at this period, we get a constant sequence and have no idea what is the actual function as shown by the graph in Figure (4.23).

A rule of thumb is to sample at $T = \frac{1}{4}$ (one quarter the natural period, in the general case). By doing this we assure ourselves of getting a more accurate representation of the signal.

The peaks from the laser data occur at fairly uniform intervals. Contained within the 9093 total data points are 1266 local maxima. Therefore we approximate the natural period to be $\frac{9093}{1266} = 7.2$ data points per cycle. We then choose $\tau = 2$. Figure

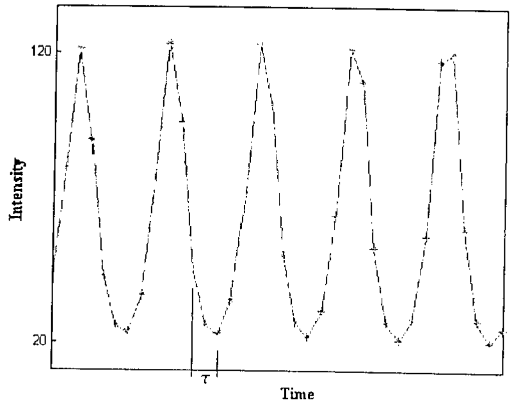

Figure 4.24: Each star represents a discrete data point of the signal. A time delay of twice this interval works well for embedding the data.

(4.24) shows the samples of the signal.

4.4.2. CHOOSING M

Once we have chosen τ we must choose M the embedding dimension. To decide the embedding dimension we use the technique of finding false near neighbors. Say we have an embedding of dimension d. Say that we have two points x and y which are far apart, i.e. they are not near neighbors in the d dimensional embedding. If, when we project x and y into dimension $d-1$, they appear to be near neighbors, we say that x and y are false near neighbors in the dimension $d-1$ embedding. The presence of many false near neighbors indicates that the data should be embedded in a higher dimension. This test requires an arbitrary measure of closeness, ϵ. If two points are closer together than ϵ in the $d-1$ dimensional embedding but farther apart than ϵ in the d dimensional embedding, they are considered to be false near neighbors. Figure (4.25) shows the graph of false near neighbors shown as a percentage of the number of pairs of points for several dimensions and $\epsilon's$.

The embedding dimension appears to be $d = 4$ or 5, since in this embedding the percentage of false near neighbors has dropped to almost zero.

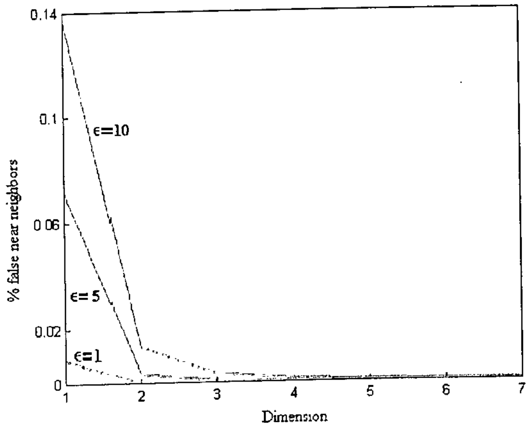

Figure 4.25: Percentage of false near neighbors of embedded laser data for various dimensions and epsilons.

4.5. WHERE TO GO NEXT

The next step is to look at predictions on the multi-dimensional case ($M > 1$). Let $\mathbf{X}(t_i)$ be the delay vector at time t_i for a given time series and embedding,

$$\mathbf{X}(t_i) = (x(t_i), x(t_i - \tau), x(t_i - 2\tau), ..., x(t_i - (M-1)\tau)).$$

We will look for the evolution rule, $\mathbf{G} : \mathbf{G}(\mathbf{X}(t_i)) = \mathbf{X}(t_{i+1})$. Say we want to approximate $\mathbf{G}$ at some particular $\mathbf{X}(t_n)$. We collect the k nearest neighbors to $\mathbf{X}(t_i)$, and regress a linear approximation for $\mathbf{G}$. Again we will use the largest k such that the sub-model is appropriate. In this notation $\mathbf{G}$ is a function from $\Re^M \to \Re^M$. $\mathbf{G}$ is a degenerate function though, since all but one component of $\mathbf{X}(t_{n+1})$ is already determined by $\mathbf{X}(t_n)$. This will simplify the computations and we only need to fit a function $\mathbf{G}' : \mathbf{G}'(\mathbf{X}(t_i)) = x(t_{i+1})$ (Bollt, 1999).

5. CONCLUSION

In this project we explored the procedure known as time delay embedding and its use in making predictions on time series generated from chaotic systems. Our goal was to determine a way to make a statement of confidence along with the prediction. We had two other constraints. We wanted to do this in an algorithmic way and we wanted the procedure to be a general one which, in principle, we could apply to any chaotic time series.

The method of prediction we used required us to embed the time series into delay space. For our purposes, we assumed that we are given a good embedding. This means that we know the time delay, and the embedding dimension for the time series. To make a prediction from a point, first we found a set of nearest neighbors to the point in delay space. We then regressed a model through these neighbors and used the model to determine the evolution of the prediction point.

We showed that if we ensured that our model adequately fit the data, then statistics would provide an accurate method of determining confidence intervals. Our question then became how to best choose k. The simplest model to fit is the linear one. A local linear approximation to an analytic function is valid only over a small region. Since this project only considered finite fixed data sets, k directly determined the size of the region the model is fit over. If k was chosen too large, the data would significantly curve away from the model. This seemed to indicate that a small k would be best. We did not want k to be too small though, since the effects of noise are more pronounced in a small data set. We showed that a way to balance these two considerations was to fit the largest k possible such that the model still sufficiently fits the data.

What does it mean for a model to adequately fit data? One meaning is that the model one degree higher actually degenerates to the original model. In the case of the linear model, we would first fit the quadratic (full) model, then examine the quadratic coefficients. If they were insignificant then the model degenerates to the linear (sub) model. In this case we say that the linear model adequately fits the data. Statistics provides a way to test these coefficients for significance. Since we then had a model which adequately fit the data, the confidence intervals of prediction were accurate.

In answering our original question concerning confidence intervals, we had to answer a more basic question concerning prediction. Both of these questions were answered using statistics which provides the algorithm based approach we were looking for. It also gives the benefit of treating the time series as a generic data set without attempting to utilize any knowledge of the system whence the data came.

6. WORKS CITED

Abarbanel, Henry D. I. *Analysis of Observed Chaotic Data.* New York: Springer-Verlag, 1996.

Abramowitz, M. and I. A. Stegun, "Handbook of Mathematical Functions", Government Printing Office, 1964, 26.6.2 (referenced in Jones).

Alligood, Kathleen T., Tim D. Sauer, and James A. Yorke. *Chaos, An Introduction to Dynamical Systems.* New York: Springer-Verlag, 1996.

Bollt, Erik. "Model Selection, Confidence, and Scaling in Predicting Chaotic Time-Series." *manuscript submitted to International Journal of Bifurcations and Chaos.*

Burden, Richard L., J. Douglas Faires, and Albert C. Reynolds. *Numerical Analysis.* Boston: Prindle, Weber, & Schmidt, 1978.

Devaney, Robert L. *An Introduction to Chaotic Dynamical Systems.* USA: Addison-Wesley Publishing Company, 1989.

Györgyi, L. and R. J. Field. *Nature* (London) **355,** 808. 1992.

Hübner, U., Weiss, C. O., Abraham, N. B. and Tang, D. (1993). Lorenz-like chaos in NH_3-FIR lasers, in [Weigend and Gershenfeld (1993)].

Jones B.A., 1-18-93, Copyright (c) 1993 by The MathWorks, Inc., Revision: 1.1, Date: 1993/05/24 18:56:29

Kantz, Holger and Thomas Schreiber. *Nonlinear Time Series Analysis.* United Kingdom: Cambridge University Press, 1997.

Neter, John, William Wasserman, and Michael H. Kutner. *Applied Linear Statistical Models, 3rd edition.* Richard D. Irwin, Inc., 1974.

Sauer, Tim. "Time Series Prediction by Using Delay Coordinate Embedding." *Time Series Prediction.* Ed. Andreas S. Weigend, and Neil A. Gershenfeld. Reading, MA: Addison-Wesley Publishing Co., 1994. 175-193.

Takens, F. "Detecting Strange Attractors in Turbulence." Dynamical Systems and Turbulence, Warwick 1980 (Coventry, 1979/1980), pp. 366-381. *Lecture Notes in Mathematics.*, 898. Springer, Berlin - New York, 1981.

Walpole, Ronald, and Raymond H. Myers. *Probability and Statistics for Engineers.* U.S.A.: The MacMillan Company, 1972.

Weigend, A. S. and Gershenfeld, N. A. (1993). *Time Series Prediction: Forecasting the Future and Understanding the past.* Santa Fe Institute Studies in the Science of Complexity, Proc. Vol. XV, Addison-Wesley, Reading, MA.